DOCUMENT RESUME

ED 071 370 EM 010 524

TITLE	Proceedings of the Canadian Symposium on Instructional Technology (Calgary, Alberta, May 24-26, 1972).
INSTITUTION	National Research Council of Canada, Ottawa (Ontario).
PUB DATE	May 72
NOTE	347p.
EDRS PRICE	MF-$0.65 HC-$13.16
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; Computer Programs; *Computers; Computer Science; Conference Reports; Cybernetics; Educational Development; Educational Technology; *Instructional Improvement; Instructional Media; *Instructional Technology; Management Education; Mathematical Logic; *Media Technology; Programed Materials; Program Evaluation; Programing Languages; Symposia
IDENTIFIERS	Canada

ABSTRACT

A Symposium was held in May of 1972 to assist in the resolution of the varied problems effecting the advance of educational technology. Most of the papers presented at the symposium dealt with the current situation of various aspects of computer assisted instruction (CAI) throughout Canada. Other papers covered CAI centralization, minicomputers, co-operative research projects, educational games, educational management simulations, logic instruction, program design, language standards, group use of CAI, teacher-authored instruction, and "do-it-yourself programing." The concluding speaker noted that a very small amount of hard data was presented in the papers and emphasized that the fate of CAI will be determined by cost-benefit analysis. (MC)

ED 071370

NATIONAL RESEARCH COUNCIL OF CANADA
CONSEIL NATIONAL DE RECHERCHES DU CANADA

PROCEEDINGS OF
THE CANADIAN SYMPOSIUM ON INSTRUCTIONAL TECHNOLOGY

COMPTES RENDUS DU
COLLOQUE CANADIEN SUR LA TECHNOLOGIE PÉDAGOGIQUE

FILMED FROM BEST AVAILABLE COPY

EM 010 524

CALGARY
MAY 24 – 26
24 AU 26 MAI
1972

NRCC No. 12726

ED 071370

CANADIAN SYMPOSIUM ON INSTRUCTIONAL TECHNOLOGY

COLLOQUE CANADIEN DE TECHNOLOGIE PÉDAGOGIQUE

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

SPONSORED BY

UNIVERSITY OF CALGARY AND

N.R.C.

ASSOCIATE COMMITTEE ON INSTRUCTIONAL TECHNOLOGY

SOUS LES AUSPICES DE

L'UNIVERSITÉ DE CALGARY ET DU

C.N.R.

COMITÉ ASSOCIÉ DE TECHNOLOGIE PÉDAGOGIQUE

This document was processed for EDRS by ERIC/EM. We are aware that some pages probably will not reproduce well. However, this is the best available copy, and we feel that the document should not be withheld from interested readers on the basis of several unreadable pages alone.

CALGARY, ALBERTA

MAY 24-26, 1972

DU 24 AU 26 MAI, 1972

TABLE OF CONTENTS
TABLE DES MATIÈRES

PAGE

INTRODUCTION 1

REGISTRANTS/PARTICIPANTS AU COLLOQUE 2

SYMPOSIUM OPENING/OUVERTURE DU COLLOQUE 11

THE NRC ASSOCIATE COMMITTEE ON INSTRUCTIONAL TECHNOLOGY AND THE DYNAMIC EDUCATION MILIEU 13

KEYNOTE ADDRESS 26

DESCRIPTION DES ACTIVITÉS CAI AU MINISTÈRE DE L'ÉDUCATION DU QUÉBEC 35

PROGRAMMES DE SUPPORT A L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR 48

MAKING C.A.I. MAKE A DIFFERENCE IN COLLEGE TEACHING 64

CAL TERMINALS - SOME RECENT CANADIAN INNOVATIONS 74

THE ROLE OF THE MINICOMPUTER IN CAI 82

A MODEL OF A CENTRALIZED CAI SYSTEM 97

LES APPLICATIONS DE L'ORDINATEUR À L'ENSEIGNEMENT À L'UNIVERSITÉ DE LOUVAIN 106

SYSTÈME INFORMATIQUE DE CONTRÔLE D'APPRENTISSAGE 142

UN OUTIL POUR L'AUTOMATISATION DES COURS: LES OBJECTIFS PÉDAGOGIQUES 173

COMPUTER ASSISTED INSTRUCTION ON A SMALL COMPUTER 187

THE N.R.C. COMPUTER AIDED LEARNING COOPERATIVE RESEARCH PROJECT 195

COMPUTER-ASSISTED MATHEMATICS FOR UPGRADING STUDENT SKILLS 206

SIMULATING AN INSTRUCTIONAL SYSTEM FOR AN EDUCATIONAL GAME 221

THE USE OF A TERMINAL BASED SIMULATION IN MANAGEMENT EDUCATION 231

TEACHING LOGIC WITH CAL USING APL 239

PAGE

THE DESIGN AND EVALUATION OF AN ADAPTIVE TEACHING SYSTEM 252

APPROACHES TO THE DESIGN AND PRODUCTION OF CAI SEQUENCES 268

LEARNING TO USE AN INTERACTIVE COMPUTING SYSTEM 283

MÉTHODE ANALYTYQUE DE CONSTRUCTION DE COURS 294

DESIGNING PROGRAMMING LANGUAGES FOR COMPUTER-BASED INSTRUCTIONAL SYSTEMS IN RELATION TO INSTRUCTIONAL PROBLEMS 323

THE USE OF APL IN COMPUTER ASSISTED LEARNING 332

CAL LANGUAGE STANDARDS 350

INTERACTIVE PROGRAMMING LANGUAGES ADAPTED FOR SPECIFIC INSTRUCTIONAL USES 360

L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR: LES PROJETS DU CERI 375

COMPUTER-AIDED DESIGN INSTRUCTION A COMBINATION OF DO-IT-YOURSELF PROGRAMMING AND COMPUTER PACKAGE USE 388

THE USE OF GROUPS IN COMPUTER-ASSISTED INSTRUCTION 395

VOCATIONAL COUNSELLING BY MINI-COMPUTER 405

THE ECONOMICS OF VARIOUS SCHEMES TO AUTOMATE FRENCH TEACHING 413

THE TEACHER-AUTHORED INSTRUCTION MANAGER (T A I M) 423

A LABORATORY FOR CAI LESSON RESEARCH & DEVELOPMENT 437

THE CAI ACTIVITY OF THE DIVISION OF EDUCATIONAL RESEARCH SERVICES THE UNIVERSITY OF ALBERTA, EDMONTON 447

ASKING THE RIGHT QUESTIONS ABOUT INSTRUCTIONAL APPLICATIONS OF THE COMPUTER 457

CLOSING SUMMARY 469

INTRODUCTION

The Canadian Symposium on Instructional Technology was held at Calgary University on May 24-26, 1972. The Symposium was sponsored by the N.R.C. Associate Committee on Instructional Technology and the University of Calgary.

Notes:

The abstracts of the papers are in front of each paper. They are in both official languages. The original is at the top of the page and the translation is at the bottom. The papers and the titles in the index are in the language in which they were presented.

The opinions expressed in the papers are those of the speakers and not necessarily those of the National Research Council or the University of Calgary.

INTRODUCTION

Le Colloque canadien sur la technologie pédagogique, tenu à l'Université de Calgary du 24 au 26 mai 1972, a eu lieu sous les auspices du Comité associé de technologie pédagogique du CNRC et l'Université de Calgary.

Notes:

Les résumés des communications, dans les deux langues officielles, se trouvent en tête de chaque communication. L'original est en haut de la page et la traduction en bas. Les communications et les titres bibliographiques sont donnés dans la langue d'origine.

Les opinions exprimées dans les communications sont celles des auteurs et ne sont pas nécessairement celles du Conseil national de recherches ou de l'Université de Calgary.

CANADIAN SYMPOSIUM ON INSTRUCTIONAL TECHNOLOGY

COLLOQUE CANADIEN DE TECHNOLOGIE PEDAGOGIQUE

May 24 - 26, 1972
du 24 au 26 mai, 1972

SYMPOSIUM CHAIRMAN
PRÉSIDENT DE LA CONFÉRENCE

TYLER, F. T.

Professor of Education
University of Victoria
Victoria, British Columbia

GUEST SPEAKERS
CONFÉRENCIERS INVITÉS

HARTLY, J. R.

Research Councils' Computer
Based Learning Project
The University
Leeds IS2 9JT
England

HUBBARD, G.

Director
National Council for Educational
Technology
160 Great Portland Street
London WI, England

JONES, A.

Professeur
Université Catholique de Louvain
Laboratoire de Physique Générale
Celestijenlaan, 200C
3030, Heverlé-Louvain
Belgique

KIRCHBERGER, A.

Centre for Educational Research
and Innovation
2, rue Andre-Pascal
Paris XVIe, France

TAYLOR, E. F.

Educational Research Center
Room 20-C228
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
U. S. A.

ZINN, K. L.

Centre for Learning and Teaching
University of Michigan
109 East Madison
Ann Arbor, Michigan 48104
U. S. A.

REGISTRANTS

PARTICIPANTS AU COLLOQUE

ANGER, Clifford D.	Department of Physics The University of Calgary Calgary 44, Alberta
BARKER, W. G.	Division of Biological Sciences The University of Manitoba Winnipeg 19, Manitoba
BEATON, Mary	Ed. C. & I. Dept. The University of Calgary Calgary 44, Alberta
BEEVOR, Walter G.	Vice-Principal Centre Hastings Secondary School Madoc, Ontario
BITTERLICH, Wolfgang	Math Department Nova Scotia Technical College P.O. Box 1000 Halifax, N.S.
BLACKMORE, W. R.	Room N. 636, Ross Building York University 4700 Keele Street Downsview, Ontario
BOND, Michael	Co-ordinator of Computer Services Room N. 815, Ross Building York University Downsview, Ontario
BOOTH, Andrew D.	President's Office Lakehead University Thunder Bay, Ontario
BOYD, Gary M.	Assistant Professor, CIT H960 Sir George Williams University Montreal 107, Quebec
BRAHAN, J. W.	Radio & Electrical Engineering Division National Research Council Montreal Road, Ottawa K1A OR8
BREBNER, Ann	Computer Applications Unit Faculty of Education University of Calgary Calgary 44, Alberta
BREBNER, M. A.	Computer Applications Unit Faculty of Education University of Calgary Calgary 44, Alberta

BRIEN, Robert	Service Informatique Min. Education 265 De La Couronne Quebec, P.Q.
BROWN, W. C.	Radio & Electrical Engineering Division National Research Council Montreal Road, Ottawa K1A OR8
BURNETT, J. Dale	Faculty of Education Queen's University Kingston, Ontario
CAREY, Peter M.	Lektromedia 88 Ashgrove Pointe Claire 710, Quebec
CARTWRIGHT, Glenn F.	Dept. of Educational Psychology University of Alberta Edmonton, Alberta
CHANNER, Carole	Department of Mathematics Vanier College 821 St. Croix Blvd. Montreal 379, Quebec
CHEVALIER, Guy	Service Informatique Min. Education Chambre 202 265 De La Couronne Quebec, P.Q.
COLPITTS, Beverley	Department of Computer Science University of Western Ontario London 72, Ontario
COLWELL, Larry L.	Computer Centre Basement, Killam Library Dalhousie University Halifax, N.S.
CRAWFORD, D. H.	Faculty of Education Queen's University Kingston, Ontario
DAYKIN, Philip N.	Computing Centre University of Lethbridge Lethbridge, Alberta
DEMPSTER, J. R. H.	Computer Science Department University of British Columbia Vancouver 8, B.C.
DOUGLAS, Sandy	Ryerson Systems Institute 50 Gould Street Toronto, Ontario
EPP, Marvin K.	Mathematics Section Northern Alberta Institute of Technology 11762-106 Street Edmonton, Alberta T5G 2RI

ERHARDT, Reg. W. D.	Southern Alberta Institute of Technology 1301-16th Ave. N.W. Calgary, Alberta
FLEMING, L. T.	Computing Centre Ottawa University 559 King Edward Avenue Ottawa, Ontario K1N 6N5
FROHLOFF, E. C.	Bell Canada 1050 Beaver Hall Hill - Room 813 Montreal, P.Q.
GALLO, Joseph G.	Department of Educational Psychology University of Calgary Calgary 44, Alberta
GARDINER, D. J.	Computing Centre University of British Columbia Vancouver 8, B.C.
GARRISON, Randy	1236-16 Ave. N.W. Calgary, Alberta
GOODMAN, H. J. A.	Ed. C.I. Department University of Calgary Calgary 44, Alberta
HALLWORTH, H. J.	Computer Applications Unit Faculty of Education University of Calgary Calgary 44, Alberta
HARASYN, Peter	8048 Huntwich Hill Calgary, Alberta
HARROP, Ronald	Professor Department of Mathematics Simon Fraser University Burnaby 2, B.C.
HART, John Francis	Computer Science Department Room 2036, Engineering Sciences Bldg. University of Western Ontario London 72, Ontario
HEBERT, G.	York University Downsview 463, Ontario
HEBERT, Marcel E.	Ecole Polytechnique 2500, ave Marie-Guyard Montreal 250, Quebec
HELM, Michael	Communications & Media Centre University of Ottawa 577 King Edward Avenue Ottawa, Ontario K1N 6N5

HILL, David R.	Division of Computing Science University of Calgary Calgary 44, Alberta
HIVON, Fernand	Université du Québec à Montreal Boîte postale 3050, succursale b Montréal 110, Québec
HUNKA, Steve	Division Educational Res. University of Alberta Edmonton, Alberta
JAMES, William	Department of Civil Engineering McMaster University Hamilton, Ontario
JONES, G. H. S.	Training Command Headquarters Canadian Forces Base Winnipeg Westwin, Manitoba R2R OTO
KAMRA, Ardis	Department of Education Province of Alberta 10105-109 St., Executive Building Edmonton, Alberta
KIEREN, Thomas E.	Dept. of Sec. Education University of Alberta Edmonton, Alberta
LAMERAND, Raymond	Centre Linguistique D'Ottawa Chemin Carson 800 Ottawa, Ontario KlA OM7
LAVOIE, Jean	Ecole Polytechnique 2500 Ave. Marie-Guyard Montreal 250, P.Q.
LEE, William	Service Informatique Min. Education 265 De La Couronne Quebec, P.Q.
LOFSTROM, M. M.	Saskatchewan Teachers' Federation Box 1108 Saskatoon, Saskatchewan
LOW, Roy A.	Community College Division 2055 Notre Dame Ave. Winnipeg R3H OJ9
LOWER, Stephen K.	Department of Chemistry Simon Fraser University Burnaby 2, B.C.
McDOUGALL, D.	3507 Benton Drive N.W. Calgary 43, Alberta

McLEAN, Robert S.	Ontario Institute for Studies in Education 252 Bloor St. W. Toronto 5, Ontario
MEINCKE, P. P. M.	Erindale College University of Toronto Mississauga, Ontario
MEUNIER, Jean Guy	Université du Québec à Montreal 1180 Bleury Montreal, Quebec
MIEDZINSKI, J.	Science Council of Canada 150 Kent Street Ottawa, Ontario K2P 5P4
MITCHELL, P. David	Director Graduate Programme in Educational Technology Department of Education Sir George Williams University Montreal 107, Quebec
MUIR, Walter	Faculty of Education University of Victoria Victoria, B.C.
OLIVIER, W. P.	Ontario Institute for Studies in Education 252 Bloor St. W. Toronto 5, Ontario
PALOIAN, Anna	Department of Computer Science Faculty of Science The University of Western Ontario London 72, Ontario
PAQUIN, Raymond	Psycho-Ped. Université du Québec 12775 Boul. Des Forges Trois-Rivieres, P.Q.
PAQUIN-ROBERGE, Mme. Luce	Service Informatique Min. Education 265 De La Couronne Quebec, P.Q.
PLANTE, Jean-Louis	Université du Québec à Rimouski 300 ave des Ursulines Rimouski, P.Q.
PROCOS, Dimitri	Asst. Prof. of Architecture Nova Scotia Technical College P.O. Box 1000 Halifax, N.S.

RETTIE, R. S.	Executive Director National Research Council Sussex Drive Ottawa, Ontario K1A OR6
ROBARDEY, Carlton P. Sr.	University of Tennessee, Martin 223 Cleveland Street Martin, Tennessee
ROID, G. H.	Centre for Learning and Development McGill University P.O. Box 6070 Montreal 101, Quebec
SANDALS, Lauran H.	#8, 933-3rd Ave. N.W. Calgary, Alberta
SCHULZ, Henry William	Department of Educational Psychology University of Calgary Calgary 44, Alberta
SHORE, Bruce M.	Centre for Learning and Development McGill University P.O. Box 6070 Montreal 101, Quebec
ST.PIERRE, Claude	Département de physique Faculté des sciences Cité Universitaire Université Laval Québec 10, Québec
STRAUGHAN, Joan Lauer	Western Washington State College Bellingham, Washington 98225
TAPPAY, A. Robert	Ontario Educational Communications Authority 2180 Yonge Street Toronto 295, Ontario
TATE, John	Lektromedia 88 Ashgrove Pointe Claire 710, Quebec
THOMAS, P. A. V.	Electrical Engineering Department University of Windsor Windsor, Ontario
TREMBLAY, Yolande	Université du Québec à Rimouski 303 St. Hubert Rimouski, P.Q.
TROSS, R. G.	Chairman Computing Centre Advisory Committee University of Ottawa Ottawa, Ontario

VASKEVITCH, David	Erindale College The University of Toronto Toronto, Ontario
WALKER, James T.	John Abbott College P.O. Box 2000 Ste. Anne de Bellevue, P.Q.
WARREN, F. G. Ross	R.C.A. Limited Trans Canada Highway, North Service Road Ste. Anne de Bellevue, P.Q.
WEIR, Jim	Northern Alberta Institute of Technology 11762-106 St. Edmonton, Alberta
WELCH, Graeme	Chemistry/Biology Program John Abbott College CEGEP P.O. Box 2000 Ste. Anne de Bellevue, P.Q.
WESTROM, Marvin L.	University of Alberta 948 Education Building Edmonton, Alberta
WHEELER, Francis M.	Assoc. Prof. of Classics Beloit College Beloit, Wisconsin U. S. A.
WHITE, E. L.	College of New Caledonia 2001 Central Avenue Prince George, B.C.
WHITWORTH, F. E.	The Saskatchewan School Trustees Association 570 Avord Tower Regina, Saskatchewan
WILLIAMS, John A. (CDR)	SSO Training Development Training Command Headquarters Westwin, Manitoba
WILLIAMS, M. R.	Division of Computing Science University of Calgary Calgary 44, Alberta
WONG, J. Y.	Assistant Director Radio & Electrical Engineering Division National Research Council Montreal Road Ottawa, Ontario K1A 0R8

WOODSWORTH, J. G.
Educational Psychology
The University of Calgary
Calgary 44, Alberta

YOUNG, Larry K.
Saskatchewan Teachers' Federation
Box 1108
Saskatoon, Saskatchewan

SYMPOSIUM OPENING

Dr. F.T. Tyler the Symposium Chairman introduced himself then introduced the three speakers; Dr. A.W. Carrothers, President and Vice Chancellor, University of Calgary, Dr. R.S. Rettie, Executive Director (External Relations), National Research Council, Dr. F.E. Whitworth, Chairman of the N.R.C. Associate Committee on Instructional Technology.

Dr. Carrothers welcomed the delegates on behalf of the University of Calgary. He recalled that at one time C.A.I. had been advertised as the panacea for rising University costs. Subsequent experience has shown that C.A.I. is not appropriate to all situations and not necessarily cheaper than alternatives. Hopefully, now that unrealistic expectations have been dispelled, serious research in the area will continue to make progress in techniques to complement the time honoured methods of lecture, laboratories and library. He hoped this symposium would contribute to that end.

Dr. R.S. Rettie welcomed the delegates on behalf of the National Research Council. He expressed regrets on behalf of the president of N.R.C., Dr. Schneider, who was unable to attend. Dr. Rettie gave a short description of the structure of the National Research Council and the Associate Committee. Dr. Rettie pointed out the need for effective communication

OUVERTURE DU COLLOQUE

Le Dr. F.T. Tyler, Président du Colloque, s'est présenté et a ensuite présenté les trois conférenciers: le Dr. A.W. Carrothers, Président et Vice-chancelier de l'Université de Calgary, le Dr. R.S. Rettie, Directeur exécutif (relations extérieures) du Conseil national de recherches, le Dr. F.E. Whitworth, Président du Comité associé de technologie pédagogique.

Le Dr. Carrothers a accueilli les délégués de la part de l'Université de Calgary. Il a rappelé que, à une certaine époque, on avait annoncé que l'EAO serait la panacée pour résoudre les problèmes financiers de plus en plus graves des universités. L'expérience a démontré, par la suite, que l'EAO ne convient pas dans tous les cas et qu'il n'est pas nécessairement meilleur marché que d'autres méthodes. Il est à souhaiter que, les espoirs peu réalistes ayant été dissipés, les recherches sérieuses dans ce domaine continueront à progresser dans les techniques qui complémentent les méthodes traditionnelles de cours, de travaux en laboratoire et de lecture. Il avait l'espoir que ce colloque tendrait vers ce but.

Le Dr. R.S. Rettie a accueilli les délégués de la part du Conseil national de recherches. Il a exprimé les regrets du Président du CNRC, le Dr. Schneider, qui n'a pas pu

between the development engineer and the ultimate user, the teacher. He also stressed the need for educators to ask the right questions, even to demand the right facilities. He pointed out that the fourth generation computer will probably be delayed until the users know what they needed and how to make the right demands on the technologists. He felt that the same remarks applied to Computer Aided Instruction. Dr. Rettie hoped that this Symposium would result in a significant contribution towards improving communication between the educators and technologists and to ensure that the right subjects are communicated.

Dr. F.E. Whitworth spoke on behalf of the Associate Committee on Instructional Technology. Dr. Whitworth's paper is contained in the next following pages.

être présent. Le Dr. Rettie a décrit brièvement la structure du Conseil national de recherches et le Comité associé. Le Dr. Rettie a attiré l'attention sur le fait qu'il est nécessaire que l'ingénieur et l'enseignant se comprennent bien. Il a aussi insisté sur le fait que les éducateurs doivent poser la bonne question et même exiger de bonnes installations. Il a fait remarquer que la quatrième génération d'ordinateurs ne sera probablement pas développée avant que les utilisateurs sachent ce dont ils ont besoin et exactement ce qu'ils doivent demander aux technologues. Il lui a semblé que la même chose est vraie en ce qui concerne l'enseignement à l'aide d'ordinateurs. Le Dr. Rettie a formulé l'espoir que les résultats du Colloque se solderont par une amélioration marquée dans les relations entre les enseignants et les technologues et par une assurance que les questions pertinentes seront bien posées et explicitées.

Le. Dr. F.E. Whitworth a parlé de la part du Comité associé de technologie pédagogique. On trouvera l'allocution et la communication du Dr. Whitworth dans les pages suivantes.

THE NRC ASSOCIATE COMMITTEE ON INSTRUCTIONAL TECHNOLOGY AND THE DYNAMIC EDUCATION MILIEU

F.E. WHITWORTH

THE NRC ASSOCIATE COMMITTEE ON INSTRUCTIONAL TECHNOLOGY AND THE DYNAMIC EDUCATION MILIEU

This paper has been written to tell you something about the National Research Council Associate Committee on Instructional Technology and the difficult problematic situation faced by it and others who wish to ensure that maximum benefit will be obtained from use of the newer media. As the Committee is composed of individuals with unique backgrounds, the ideas expressed herein are those of the author and do not necessarily coincide with those of other members of the Committee or of the NRC.

Because of the rapidly changing state of scientific development in this area and an unprecedented implementation of social change, any attempt to clearly identify or delimit current instructional problems, or to suggest solutions which would harness instructional technology to dynamic educational objectives is obviously doomed to some degree of failure. Yet we have no choice as to whether or not to make such efforts. To incite us to greater endeavour are some encouraging results, some extravagent predictions, and some rash, though not impossible, dreams of the computer's capabilities. These relate to revolutionary functions of computers in education which lie not in business and management applications but in the novel idea of computer-assisted learning or computer-managed instruction.

The single most powerful argument put forward in support of CAL is that it will provide for individual differences, making it possible for each subject to achieve his full potential. A second or correlative advantage refers particularly to current drill and practice programs which can be adjusted readily to individual needs. But mostly there are limits to the branches available; and the chief adjustment relates to speed of progress through the materials. A third claim relates to immediate feedback to pupil and teacher. But again, evidence at present suggests that claims here may be greatly exaggerated. A fourth relates to release of teacher time from drill activities and the marking of papers. Here again experience suggests that there is little actual saving, and there may be cheaper ways of conducting drill and practice which are equally effective.

Contrariwise there are arguments suggesting that computerized and related programs will never be adopted. They recognize what we are prone to call inertia and apathy on the part of many educators, parents and administrators, and an awareness that progress in instruction has generally been slow. The majority of band-wagon panaceas that have burst on the scene like skyrockets have fallen into innocuous desuetude shortly after. Institutional rigidity combined with technology in its infancy can hardly guarantee rapidity in adoption and usage. Essentially, few schools are ready for computer systems and few computer materials now available are ready to contribute much to the schools.

Yet there is a growing awareness that the wants facing education are not being satisfactorily provided for. Some of these anxieties are clearly identified and focused on specified requirements and objectives. Others reflect unease and frustration. The pressures for change will not be ignored. Educational planning must involve all aspects of society and resolve conflicting demands. Such planning must be an ongoing function. Instructional technology has an important place in this and an important aspect of it is the use of the computer. It is therefore appropriate and vital for educationists, provincial, national and international to consider and suggest proper approaches and procedures for its development in the field of education.

This symposium is but one of the sort of attempts which will have to be undertaken if we are to solve some of the problems facing us related to effecting advance in instructional technology. In scope this program has been limited to considerations of using the computer in education, which appears as a wide spectrum of possibilites ranging from housekeeping to engineered instruction. Some of these relate to the planning and administration of educational services, others to the use of the computer which provides extensive calculating facilities for educational research; and still others consider the computer's possible use in providing resources for the educational process. The latter category can include information storage and retrieval, the generation of dynamic demonstrations and displays, the incorporation of evaluation and assessment devices to provide immediate feedback, provision for interaction with the program, and the possible incorporation of all of these in systems or as units in the instruction-learning process. Our concerns at these meetings will relate mainly to the computer as a learning resource, to interactive methods of instruction, and the providing of assurances that attempts to harness the computer in instruction will be attacked reasonably systematically, cooperatively and on a grand enough scale to ensure a fair hope of success.

The Associate Committee

The Associate Committee on Instructional Technology is one of a number of such committees appointed by the NRC to act as advisory panels in selected areas; to provide liaison between appropriate sections of the NRC and outside bodies concerned with advance in these areas; to advise what research should be undertaken, by whom and the priority which approved projects deserve within the constraints of expertise and money available, and with cognizance of current needs.

To committee members, applications of the computer in the instructional learning process though of prime importance is but one of their interests. It is recognized that T.V., cassettes, radio and all other media hardware can be utilized in a variety of ways, sometimes as separate resources and sometimes incorporated into a system or systems. The system may also incorporate the teacher, book and blackboard either as an integral part of a closed system, as auxiliary contributors, or as optional resources in computer-managed instruction.

The role to be given to the teacher in this is all important. A system could be provided which would operate without a teacher as has been attempted through correspondence courses, "teaching machine" programs and workbooks, I.P.I., etc. all of which have lessened or altered her role. Again the teacher could be in charge, using the media as aids; or she may act as operator of the system in a position to make alterations or changes in materials or program as desired.

Similarly, determining a suitable role for the pupil is of paramount importance. One possibility is to expect the pupil to adapt to the programs provided, essentially using equipment devised for other purposes. This is generally frowned on. A second is to provide a series of units which will be completed according to the acumen and motivation of the student. Here the student, teacher or program developer may have chosen the units the student will work through and set criteria for successful completion of each.

Membership of the Committee

In the formation of the Committee selection was made from among competent individuals, rather than from representatives of any professional bodies, though with consideration for making the committee strong and well-rounded and with some measure of geographic representation. Some requisites for an invitation were a lively interest in at least some of the branches of the growing technology, a willingness to devote time and energy to further the development of such technology for the good of education, and the wish to weigh seriously possible strategies aimed at the timely and expeditious implementation of the newer media in the instructional fields. As a result, the Committee members represent competencies in such fields or areas, as: psychology and sociology; general, adult and university education; the systems approach; engineering; communications; research design; development strategies; and business expertise from some technologically-oriented industries. Membership is normally for three years, one-third being appointed each year. Members serve without salary. They may be reappointed.

Operations of the Committee

The Committee has met three times a year during the first two years to get it firmly established, but meetings may be reduced to two in number annually henceforth. To help those interested from outside keep in touch the committee issues a newsletter at irregular intervals, which is available on request. Short contributions for publication in the newsletter are welcomed.

This Committee was thus conceived as a coordinating body in the field of instructional technology acting as an advisory body national in scope. To carry out its responsibilities it has attempted to make some advance in certain areas through organizing sub-committees which report back to the full committee.

At its inaugural meeting the Committee agreed on the following objectives:

1) To bring together specialists from the various disciplines involved in the field of instructional technology, in order to formulate requirements and establish guidelines for research and development;
2) To promote interchange of information among workers in different disciplines, as for example, through the organization of symposia;
3) To promote and coordinate relevant research and development in universities, government laboratories, and industry, with the aim of establishing a strong, Canadian-based capability in this field;
4) To encourage a degree of standardization in instructional systems sufficient to permit all provinces to benefit from the inter-

change of instructional material.

Sub-Committees

It has been noted that the Committee aims to bring together specialists from various disciplines combining the theoretical and practical in formulating guidelines for appropriate research and developmental endeavour. Its sub-committees are appointed from among its members and experts outside the Committee to undertake specific tasks. These sub-committees complement other efforts of the Committee which seeks to accomplish the setting out and pin-pointing of endeavour through calling seminars or workshops designed to provide for a meeting of minds and exchange of ideas. Such meetings can introduce international, national and local content and help to ensure advance which is most likely when undertaken at not one but all levels.

One of the sub-committees constituted to date has been charged with developing specifications for Computer Terminals for Instructional Applications. It was considered that there was an urgency here and that therefore this task should receive high priority. Adoption of minimum specifications could be beneficial to buyers, educators and manufacturers. Detailed engineering and functional as well as systems performance specifications need to be set out. Standard features and options need to be listed with consideration for noise level, appearance, feel, portability, reliability, display rate, line length and other pertinent specifications.

A second sub-committee was appointed to deal with the sharing of instructional technology materials. It was recognized that there is need to provide for the easy and efficient distribution of materials if we are to keep costs down, eliminate needless duplication of efforts and produce a quantity of quality programs in a reasonable time. There are difficult problems of costs and copyright but answers to these must be found. In addition, in Canada materials must be available in the two languages, English and French.

The difficulty of providing networks for the distribution of program materials is impressive. The communication tools may include satellites, cables, wire and wireless, cassettes, films, records, etc. How these are to be used may depend on whether representation is one, two or three dimensional, whether communication is on line in the interactive mode between teacher and pupil, which peripherals of the computer will be used, and the percentage of the process to be taken over by computerized education.

The problems mentioned, and there are many more, are of concern to other bodies than the Committee, such as the Canadian Standards Association, the Department of Industry, manufacturers, Departments of Education, and in fact, anyone interested in marketing or using related materials.

A mail survey of the field led the Committee to the conclusion that there is a woeful lack of reliable information on currently available materials. This problem is by no means unique. Years ago a similar one was faced in the area of tests and measurements and resulted in the preparation in the U.S.A. of a yearbook covering specifications on the bulk of standardized tests. In addition it appears that much material has only limited local application. Some of it has been lifted bodily despite copyright restrictions, and some is restricted in distribution because of local regulations. Much of it would not be generally acceptable since most people today expect a high commercial quality of the materials on sale.

A third sub-committee is a working panel concerned with computer languages. The advantages of having a standard language with wide usage and provision for modification as needed, and one having accented French characters seems patently clear for Canada. Only manufacturers interested in cornering the market, or a large part of it, or lone wolf researchers, might disagree. This sub-committee has recognized that materials are currently appearing in many languages and suggests that efforts to achieve agreement on one language should be given a high priority.

The Committee's Commitment to Research

The Committee is dutybound to attempt the following: to identify specific areas concerning use of automated instruction which require research and development; assess the research activities needed and already completed; advise on the facilities and manpower required to complete the others; provide an estimate of costs for each; make recommendations as to where each could be carried out best and by what body e.g. industry, university, government, other facility, etc; and establish a set of priorities to cover the projects. One sub-committee is concerned with research needed and priorities.

Before we can consider the implementation of computer applications and make use of the newer media to the extent that automation has affected other industries, educators must first identify the purposes of education and establish ultimate, intermediate and behavioral goals. This need not go so far as to provide one or several systems, or the adoption of a behavioristic approach. Education will always be an art and its product unique pieces, but there is a place for science in it. The role of the teacher will change, but only for him to become more professional.

Within the constraints of money, professional competence and communication expertise, such identified problems should be attacked and solutions sought through research. This suggests a need for better planning in our research and development undertakings. We recognize that progress has and will continue to result from sporadic efforts of individuals in scattered laboratories working on an inspirational and creative basis. The bulk of research today, however, is conducted on a concerted systematic base. As much of this research is costly, interdisciplinary, long-term, and demanding support can be sought from only a few sources, and normally only when there is assurance that a well conceived plan will be executed expeditiously, efficiently and with some chance for success.

Research should be directed to the testing and evaluation of software and hardware and this research should be conducted at all levels from local to international. Just as research should be undertaken at all levels so is there need for the establishment of standards which are generally approved. This is necessary to effect maximum transferability of hardware and software systems and materials. International cooperation can help to ensure coherent national programs and international communication lines, providing for joint provincial, national and international projects.

One concern throughout relates to compatibility, or the ability of separate systems elements to be interconnected and work together, whether we are considering hardware components, software, or the media or character codes. Limiting factors may be the speed of operation e.g. a typewriter keyboard; electrical, e.g. the voltage of signals; load, etc.; logical, such as code compatibility and control logic, and control signals, etc. There are conversion devices which will link incompatible materials in some instances but one should be sure that the linkage is satisfactory. At best it is a poor substitute for providing compatible component parts in the first place.

The Current Situation

It follows from what has been said that the Committee, as with others working in the field, cannot work without full cognizance of the current educational context. Today we have many enigmatic situations in society. In education many persons continue to favour established traditional methods which use labour intensive procedures. They resist technological innovations as dehumanizing, depersonalizing and destructive of teacher-pupil rapport and the professional autonomy of teachers. Some educators have accused industry of having sold them down the river with teaching machines and programmed instruction, language laboratories, television, and more recently, contracts. At the same time there is criticism of education's inadequacies as being stodgy, old-fashioned, rutted, indecisive, providing a stereotyped program that has little relevance to modern needs, and in fact is damaging to creativity and growth for many. Researchers provide little help because of conflicting evidence as to the relative value of book, human voice, the talking face, films, television, radio, etc. Education today is no place for timid souls.

Instructional technology emerged in 1950 hopefully to provide individualized instruction and an enriched program. Teacher acceptance has been varied and it appears that teachers most readily accept those materials that are simple in operation, reliable, and least affect their traditional role. There is need for an early beginning in exposing teacher trainees and teachers to the newer media so that they will feel at home with them and appreciate their possibilities.

An evaluation of the present state of development in instructional technology, and efforts to make this known to teachers, parents, school boards, and the general public is essential if we are neither to oversell nor undersell the true situation and hurt the chances of eventual large scale adoption. Implementing new approaches is always difficult and often unsuccessful. There is no assurance that all promising proposals for innovation will be given a fair trial or that we shall benefit from them as much as we should. One thing that complicates the situation and increases the obstacles in the way of modernizing education through the use of the newer media is the lack of agreement on method of attack and the goals to be achieved. The whole problem must be attacked from many sides if success is to be achieved.

Current literature, among other things, suggests that education today must meet the needs of all persons from infancy to old age. It suggests that it must be relevant. Some would have it pupil-centred to a degree where each student would select his own course, units of work and study approaches. Others would provide stereotyped courses and still others favour the traditional approach covering a variety of procedures. Some are sure that the use of the newer media goes hand in hand with a narrow behavioristic approach and that both teacher and pupil will be relegated to the position of objects in an on-going process concerned with input and output only. There is also some confusion as to whether or not we are talking in terms of one, or a few large systems, or of a number of unrelated small systems, rather than both large and small computer utilization with access among them and with provision for incorporating the older as well as the newer media into the education process. There is some concern as to whether the computer will take over education or whether we are talking in terms of media as education aids. The relationships between learning theory, curriculum construction, and the preparation of units remains unclear.

The choice of materials at present is extensive and varied with equipment being portable or fixed and programs piped in from central stations, or initiated on the spot, etc. However the software available is decidedly limited in quantity and much work is needed here.

To date systems have either been explored as demonstration or development devices with primary emphasis on determining feasibility of greater development of the technology. Some work has been done on the development of educational content, its distribution and operation, with little research on the basic teaching-learning process made possible by interactive computer-based instructional systems. The gap between the preparation of materials and their distribution and the incorporation of these into current instructional procedures must be subjected to research which shows promise of resulting in the modification of both the materials and programs prepared, and of the conventional methods of instruction.

It appears imperative that the education community take the initiative in accepting responsibility for defining goals and uses of computer technology in education. The functions performed by the computer and the materials to be developed for use in instruction should be the unique responsibility of education.

At the same time there is need for intercommunications among industry, education and government personnel with each involved and contributing according to his responsibilities and making possible contributions in hardware, software, distribution systems, pilot projects and evaluation. Such intercommunication cannot help but be mutually beneficial. Without it, and without planning, no one can predict the future course of instruction in Canada. At least this is how one member of the Committee sees the current situation, as one in which the challenge is to have specialists in various disciplines related to instructional technology, formulate requirements and guidelines, promote interchange of information, foster and provide for the coordination of research and seek to achieve the degrees of standardization that appear necessary.

KEYNOTE ADDRESS

by

Geoffrey Hubbard

Director

United Kingdom National Council for Educational Technology by

Geoffrey Hubbard

Director

United Kingdom National Council for Educational Technology

May I first address myself to the question, "What is educational technology?" Strangely enough, we haven't got that one sorted out yet; quite often I hear people expressing regrets that the term was ever introduced, and wishing we could find another, better description. But nobody ever comes up with one, and without wishing to sound like an old China Hand, I do remember, a mere five or six years ago, what a relief it was to stop heading one's memoranda "The use of language laboratories, audio visual techniques, closed circuit television, programmed learning and so on in education" and to settle for the comprehensive phrase, "educational technology". If in the intervening years we haven't made it clear what the phrase means, and if it hasn't acquired a good lustre, it's not the fault of the words.

As an example of the range of meaning the term is held to cover, perhaps I can read a short passage from Chapter 2 of the recent report, "Central Arrangements for promoting Educational Technology in the United Kingdom".[1]

> "8. Educational technology comprehends a number of distinguishable areas of activity. The most familiar is the use of technical devices to support the processes of teaching and learning. These include visual projection apparatus, radio and television systems, tape recorders for sound and vision, duplicating, photographing and other reprographic equipment, language laboratories and teaching machines, from the very simple to the highly elaborate, some of which require staff with special training or experience to operate and maintain them.
>
> 9. However, the use of technical aids is not self sufficient. They are devices for conveying learning material which has to be supplied either by the individual teacher or by some other teacher or author on his behalf. Other aspects of educational technology relate therefore

/to the ...

> to the production of this material. Sometimes it stems from the general interaction of teacher and student with a range of problems and situations. At other times, it is necessary to construct the material more systematically in the light of research into the processes of learning, particularly of learning by a carefully designed sequence of steps. This process is characteristic of what has become known as programmed learning, although even here, current practices frequently embrace course structures considerably broader than those originally conceived. Thus the teacher constantly requires facilities to make resource material for himself, or to adapt to his own needs material supplied from other sources. These needs will vary from time to time, perhaps in response to changing local circumstances or to the evolution of new attitudes and approaches to learning.
>
> 10. Moreover, in responding to his day-to-day problems, the teacher who decides to integrate new systems and techniques into his work will find that his innovation has implications beyond the confines of his own classroom; it may impinge on the work of his colleagues or create new demands on time, accommodation and financial resources. The more obvious material aspects of educational technology cannot sensibly be dissociated from consideration of organisation and management, or curriculum content, innovation and development."

and, a little later in the same chapter:

> "12. ... We find general agreement that it comprehends both technical aids and resource material. Its claim to serve educational planning, organisation, building design and similar subjects has had, until now, less scrutiny and consequently less support. ..."

On the one hand it's just audio visual aids writ large; on the other it's so broad it almost comes to anything that improves the quality of education. To clarify this confusion, I suggest we look a little more carefully at the origins of educational technology. We find the dichotomy implied in the history; on the one hand a long record of application of media in education; on the other a number of conceptual innovations: the various adventures in self-pacing recorded by L.C. Taylor in his book, "Resources for Learning"[2] and the complex of ideas about learning which were the basis of the programmed learning approach.

It is, when it comes to the point, ideas which matter, and the particular ideas which have had so much influence on the development of educational technology are the ideas that in the end you cannot teach anybody anything; you can only help him to learn, and the related idea that, by defining what we want to have learned, putting a great deal of thought into the creation of a learning system, measuring how well it works and revising it in the light of that evidence, we can move towards more effective learning.

However, there is one trifling disadvantage about this idea - that the process involves a great deal of effort, a great deal more time and effort than is required for more orthodox lecturing or teaching. It is here that the two streams converge, for the way to get an adequate return from the greater investment in developing an effective systematic learning system is by applying it widely. Just as the effort involved in turning back-of-envelope lecture notes into a properly organised book was rendered worthwhile by Mr Gutenburg and his moveable types, so the further effort of developing learning systems using print and non-print materials is justified by our ability to replicate them and, by the use of modern communications systems, to make them widely available.

Thus it seems to me that educational technology is essentially a matter of concepts and ideas, but those concepts and ideas are only practicable if we use to the full the capacity to replicate and disseminate which the second phase of the industrial revolution, the communications revolution, has put at our disposal.

And how far has this interpretation of educational technology spread in the UK? Well, first of all the UK is a country where the concept of educational autonomy and decentralisation is either a religious principle or a disease, so that we have a great diversity of experiment, mostly on a rather small scale. We can show examples of almost everything - you name it, we've got it; though I wouldn't be sure of putting my hands on it in a hurry. Indeed, one of the problems in Britain is knowing just what is going on in its 35,000 schools controlled by about 200 local authorities.

Because we need this kind of information we recently asked the BBC Education Correspondent, acting as our part-time consultant, to gather together details of some of the more important innovations at school level that one doesn't normally hear about. The interesting thing is that he has been overwhelmed with reports. It's too early to generalise his findings, but it's already obvious that many more schools than we supposed are moving from technology in education to the technology of education. In some areas the developments are quite considerable. We are now going to select some of the more significant cases and document them on audio-tape, slides and films, because our experience is that teachers and administrators are much more likely to be encouraged to innovate by honest illustrations of what is already happening than by any amount of theorising and sermonising. (This is a fact we ignore at our peril!)

/At the...

At the same time, if we are only just discovering the ferment of small scale innovation, we also have some major developments which make the British scene particularly interesting - the Open University, a very significant large scale development in providing learning systems for adult students; considerable investment in closed circuit television applied to all levels of education, including schools services in two very large cities and two small cities; a wide ranging provision of educational programmes of very high quality on the public service broadcasting networks; and the wide range of curriculum studies being undertaken by the Schools Council for the Curriculum and Examinations, many of which lead to the development and publication of materials in book and non-book forms.

A particularly interesting feature has been the rapid development of interest in "resource based" learning, partially perhaps a reaction against the more rigid behaviourists and exponents of the tightly structured system. There is a tendency to equate structuring with certain subject areas - to assume that maths and science are sequential and can therefore be structured but that social science and arts subjects are not sequential and therefore accord better with a resource based approach. My own conviction is that it is not subjects as such but learning tasks which are sequential. It may well be that structuring can be applied to parts of any subject, but to the whole of none. I believe in fact that we apply much effort and energy to deciding on the media we should use (and that's a relatively simple choice, determined by an algorithm that gives the cheapest effective medium to meet a particular educational specification) while taking little account of the factors which lead to the choice of mode of learning, factors which are partly concerned with our educational aims and partly with our personal and temperamental inclinations. In the present state of knowledge, when so much educational research turns in "no significant difference" results, our learning theories are probably a much less important influence on our choices than our personal predilections.

In a country like Britain, a national agency such as the National Council for Educational Technology cannot expect to pursue a policy of centralised innovation; our activities have to be directed rather to facilitating the developments which potential innovators in the educational system are themselves anxious to implement. We look for the barriers to innovation, the road blocks which stand in people's way, and we try to remove them. We have launched some projects which produce materials or learning systems: the Continuing Mathematics Project, developing self-instructional materials for non-specialist sixth-formers; the Primary Extension Programme concerned with audio visual media and the disadvantaged child; and the development of materials to assist with in-service training in educational technology, for example. But much of our concern is with areas of activity which do not involve development of materials: with identifying

/spontaneous...

spontaneous innovations, from which we can draw case studies to guide and inform others; with creating a system to facilitate the application of educational technology in pre-service training of teachers; with the development of information systems, bibliographic techniques for non-book materials and consultancy services; and with the cracking of such specific obstacles as copyright and copying services, timetabling and scheduling, specification and standardisation.

Now despite this general strategic approach, in a far-sighted action some five years back, the Council embarked on a study which was described as an Advanced Project in Educational Technology, the idea being as general as that, that in among all the practical short-term objectives, there ought to be something that set the sights a little higher. APET made its concern the potential of the computer in education. A study team surveyed the United States and the United Kingdom scenes, with a general look at the rest of the world; a conference was held; a number of reports were published and in 1969 a little pamphlet called "Computer Based Learning: a programme for action" was put to our Government. It suggested that roughly $5.2M should be spent, over about five years, on supporting a co-ordinated programme to develop Computer Applications in Education, based on four or five centres where work was already in hand, even if on a limited scale.

I had no thought, when I accepted your kind invitation, that I would have anything further to say on that subject. In fact, just before I left the United Kingdom an announcement was made that just such a programme is to be launched, and the National Council will be closely concerned with the management of it. It will not be the programme we were thinking of in 1969, naturally, and I thought it might interest you if I shared my thoughts with you about the way in which our programme might develop. I say might, because being Director of the National Council is very like being the father of a child. You are in at the conception, in at the birth, you get a fair share of all the slog of bringing it up, but when it gets to the stage of being a fully autonomous individual, your duty is to stand back, let go, and watch it make its own mistakes. So just at this stage I can say how I shall try to help the programme to develop, what I think the priorities should be. But what actually happens is probably another story.

You probably read the 1969 reports.[3] The main emphasis was on the computer in a direct teaching, tutorial role and in a back-up management role, in fairly highly structured subjects like maths, science, medicine, or technician training, and there was particular provision for the development of student terminals, seen as providing facilities for visual presentation and something other than typed inputs. One point made there still seems sound to us; that initially CAI in any form is going to be most appropriate to the upper secondary and tertiary sectors. I don't think it matters

/very much...

very much whether our costings are accurate; it is simply that the first applications are going to be expensive; like every new application the costs will go down with time and breadth of application. Now the unit costs of education rise with the age of the student, because traditionally the pupil/teacher ratio decreases with student age. So the break-even point for the computer comes sooner at the upper end.

But some of the approaches to computers in education which have been suggested and tentatively explored do seem open to question. For example, the idea of mapping the structure of knowledge, reading it into the computer and programming a chatty informal tutorial exploration, seems both over-optimistic and educationally naive. For one thing, does knowledge have a structure? Surely not in any unique sense; there are almost an infinity of structures of knowledge, depending on what you want to use it for, how much you know to start with, what sort of attitudes you have. And the other half of the process, dressing up the computer to behave like an Oxbridge don, seems to me to be as misguided as putting false eye-lashes on the lens of a television camera.

No, we shall not get the best out of the computer by attempting to mould it into our own image. The tutorial is a peculiarly human and humane mode of instruction; perhaps it's best left to human beings. Instead I think we should look at the totality of the educational process and at the characteristics of the computer, and see where the two match. There should be a close consonance between our programme for promoting and encouraging educational technology and our programme for the computer as an educational resource.

For one thing, we are already using computers in certain roles. They are an essential element in any comprehensive information system; the only practical tool that enables you to undertake multi-co-ordinate search and retrieval of a large data base. They offer a prospect of fast, efficient scheduling - and scheduling leads to resource allocation, and resource allocation to computer managed instruction, where we already have some experience. Then again, the computer as a data processing device is almost essential to any widespread application of progressive test and assessment, particularly as a diagnostic device or a means of determining learner profiles. And the same type of diagnostic procedure, coupled with the resources of an information system, can be applied on the teacher's behalf to the analysis of educational problems and the proffering of possible contributions to their solution.

Equally, the computational power of the computer can be used to generate examples, particularly multi-variant examples, as well as providing a straightforward support service, allowing the student to attack problems which would otherwise be insoluble simply by

/doing...

doing calculations for him. Beyond this, the most difficult and challenging application is to genuine problem solving.

My own vision of the future, if our programme is in any sense successful, is thus a long way from the picture of earnest students, each in his carrell, signing on to the computer which, after a warm greeting and a friendly enquiry as to their personal well-being, moves into a Socrato-electronic dialogue. Rather I would see the computer in the classroom as the student and the teacher's Man Friday, versatile, patient, untiring, infinitely exploitable; being used at one moment to search out information, at another to keep or analyse records, then to calculate, then to generate a sequence of examples. And let us not forget that, by the time we get anywhere near this stage in education, there will be many many things we shall be able to do given only we have a terminal and a telephone line. After all, the one job we could do tomorrow - or rather yesterday - is using the computer to book for the fifth form to go to see Macbeth.

I hope that the close association of the National Council with the British Computer Assisted Learning Programme will ensure that the Programme develops with our general educational technology activity in the closest possible interaction. The aims after all are the same - to apply new concepts and techniques to the purposes of education so as to improve the efficiency and effectiveness of the process and the pleasure and enthusiasm of students and teachers. For education is about people, and we, as educational technologists in general or Computer Based Learning experts in particular, share in the responsibility for ensuring that people come first.

References

1. "Central Arrangements for promoting Educational Technology in the United Kingdom", Report of the Working Party appointed by The Secretary of State for Education and Science, published by Her Majesty's Stationery Office, London, 1972

2. "Resources for Learning", by L.C. Taylor, published by Penguin Education Special 1971

/3. "Computer...

3. "Computer Based Learning Systems: A programme for research and development", published by NCET 1969

"Computer Based Learning: a programme for action" published by NCET 1969

"Computers for Education", NCET Working Paper Number 1, published by Councils and Education Press limited in 1969

DESCRIPTION DES ACTIVITES CAI AU MINISTERE DE L'EDUCATION DU QUEBEC

William Lee
Service de l'Informatique
Ministère de l'Education
du Québec.

DESCRIPTION DES ACTIVITÉS CAI AU MINISTÈRE DE L'ÉDUCATION DU QUÉBEC

SESSION 1A
ROOM/SALLE 148

By/Par: William Lee
Directeur adjoint
Service de l'Informatique

Le Laboratoire de Pédagogie Informatique du Ministère de l'Éducation, en opération depuis le début 1970, groupe actuellement 20 personnes (11 pédagogues et 9 informaticiens) et dispose d'un matériel électronique IBM 1500 doté de 6 stations d'élèves.

Les objectifs du Laboratoire sont:

- De former un premier groupe de spécialistes en enseignement assisté par ordinateur,
- De développer des cours expérimentaux et les évaluer,
- De faire des recommandations concernant l'applicabilité du CAI dans le sens d'un développement rationnel au Québec.

Plusieurs cours ont été développés au Laboratoire en informatique, mathématiques, électronique, grammaire française et géographie. Les cours d'informatique tels que PL/1 et COBOL servent dans le cadre du système de formation technique du Service de l'Informatique. Une méthode analytique de fabrication de cours ainsi qu'un ensemble de fonctions de support ont été développés au Laboratoire.

Une expérimentation pédagogique à l'aide du système conversationnel ITF est également en cours dans plusieurs collèges du Québec. Nous avons l'intention de commencer l'utilisation du système coursewriter III vers septembre 1972.

DESCRIPTION OF THE ACTIVITIES IN CAI AT THE DEPARTMENT OF EDUCATION OF QUEBEC.

The Instructional Data-Processing Laboratory of the Department of Education, which has been in operation since the beginning of 1970, presently has a staff of 20 people (11 teachers and 9 computer specialists) and has at its disposal an IBM 1500 computer with 6 student stations.

The aims of the Laboratory are;

- To form a highly qualified group of specialists in computer-assisted instruction
- To develop experimental courses and evaluate them
- To make recommendations concerning the applicability of CAI for rational development in Quebec.

Several courses have been developed in the data-processing Laboratory: Mathematics, Electronics, French Grammar and Geography. Programming courses such as PL/1 and COBOL are used as part of technical training in the Data-Processing Service. An analytical method of course construction as well as a set of supporting functions have been developed in the Laboratory.

A teaching experiment using the ITF conversational system is under way in several Quebec colleges. It is planned to begin using the Coursewriter III system in about September 1972.

PROGRAMMES DE SUPPORT A L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR

LUCE PAQUIN ROBERGE
Laboratoire de Pédagogie Informatique
Ministère de l'Education du Québec

PROGRAMMES DE SUPPORT POUR LE CAI DEVELOPPÉS AU LABORATOIRE DE PÉDAGOGIE INFORMATIQUE

SESSION 1B
ROOM/SALLE 148

By/Par: Luce Paquin Roberge,
Analyste,
Service de l'Informatique

À partir des possibilités du système IBM 1500 et du langage coursewriter II, le Laboratoire de Pédagogie Informatique s'était fixé des objectifs quant à la souplesse des cours qui seraient produits. On a voulu entre autres:

- Donner une certaine liberté à l'étudiant en ce qui regarde la formulation de ses réponses; ceci implique une analyse des réponses plus sophistiquée que celle offerte par CWII;
- Permettre à l'étudiant de poser des questions auxquelles on fournirait une réponse adéquate le plus souvent possible;
- Qu'une notion, une fois programmée, puisse être utilisée dans plusieurs cours, s'il y a lieu;
- Avoir la possibilité, pendant le déroulement d'une leçon, d'éliminer certaines parties du cours ou d'en ajouter si nécessaire à la bonne compréhension de la suite;
- Enrichir le CWII dans le sens des besoins exprimés par les professeurs du laboratoire.

Ces objectifs nous ont amenés à parler:

- de modules,
- d'enchaînement de modules,
- de modification dans la présentation des modules,
- d'analyse de réponses,
- d'analyseur des questions d'étudiants.

Pour ce, il a fallu établir des règles d'utilisation du CWII, écrire des fonctions et opérer certaines modifications au "software" du système 1500.

A l'aide de l'expérience acquise sur le système 1500, nous avons commencé le développement d'un ensemble de fonctions susceptibles d'augmenter considérablement les possibilités de CWIII.

SUPPORT PROGRAMS FOR CAI DEVELOPED IN THE LABORATOIRE DE PÉDAGOGIE INFORMATIQUE

Taking into account the possibilities of the IBM 1500 system and the Coursewriter II language, the Laboratoire de Pédagogie Informatique set out its objectives concerning the flexibility of the courses which would be produced. It was desirable among other things:

- To give a certain freedom to the student concerning the formation of his answers; this implies a more sophisticated analysis of the answers than that offered by CWII;
- To allow the student to ask questions to which an adequate answer would be given as often as possible;
- That an idea, having been programmed once, might be used in several courses as required;
- To have the possibility, during the lesson, of eliminating certain parts of the course or of adding to it, if this is necessary for a clear understanding of what follows;
- To enrich CWII according to the needs felt by those teaching in the laboratory.

These aims have led us to speak of:

- modules,
- linking of modules,
- modifications in the presentation of modules,
- response analysis,
- students' question analysis

For this, it has been necessary to set up rules for the use of CWII, define functions and carry out certain modifications of the software of the 1500 system.

With the help of experience acquired on the 1500 system, we have started to develop a set of functions which could lead to a considerable increase in the possibilities of CWIII.

MAKING C.A.I. MAKE A DIFFERENCE IN COLLEGE TEACHING

Stephen K. Lower

Department of Chemistry
Simon Fraser University
Burnaby 2, B. C.

MAKING C.A.I. MAKE A DIFFERENCE IN COLLEGE TEACHING

SESSION 1C
SALLE/ROOM 148

By/Par: Stephen K. Lower
de l'Université Simon Fraser
Simon Fraser University

This presentation attempts to discuss some of the factors that enable a computer-based instructional system to function as part of a practical and effective learning system, operating within the academic mainstream of a university environment. It is the author's view that all too many university c.a.i. installations are so far removed from this context that c.a.i. as an instructional tool is already in some danger of following the "programmed instruction" of an earlier era down the dusty road to well-deserved oblivion.

A brief summary of c.a.i. activities at Simon Fraser University will be given, with particular mention of the organization of c.a.i. services, the use of our courseware by other institutions, and our attempts to interface audio-visual devices such as slides, audiotapes, and the Philips "P.I.P." unit to a conventional c.a.i. terminal system.

Finally, we will comment on our experience with the Coursewriter III system, and on the steps that must be taken if this system is to become a versatile and adequate instructional tool.

COMMENT AMÉLIORER L'ENSEIGNEMENT UNIVERSITAIRE À L'AIDE D'ORDINATEURS

Nous essayons d'examiner un nombre de facteurs permettant à un système pédagogique, basé sur un ordinateur, de constituer un système d'instruction pratique et efficace dans le milieu de l'enseignement universitaire. L'auteur est de l'avis que beaucoup trop d'installations universitaires sont tellement loin de ce contexte que l'ordinateur, comme moyen pédagogique d'instruction, risque déjà de suivre le même chemin que "l'enseignement programmé" d'autrefois et de sombrer dans un oubli bien mérité.

Nous donnerons un bref résumé des travaux en CAI à l'Université Simon Fraser avec mention spéciale de l'organisation des services du CAI, l'utilisation de nos systèmes de cours par d'autres établissements et nos tentatives pour brancher des dispositifs audio-visuels tels que les diapositives, les bandes sonores et le système "P.I.P." de Philips, sur les terminaux d'un système EAO.

Enfin, nous ferons part de l'expérience que nous avons acquise avec le système Coursewriter III et de ce qu'il faut faire pour que ce système devienne un outil d'enseignement satisfaisant et de grande souplesse.

MAKING C.A.I. MAKE A DIFFERENCE IN COLLEGE TEACHING

The potential benefits that direct on-line interactive c.a.i. can bring to the learning process have been pointed out by numerous authors; that few of these benefits have yet been realized in practice at most institutions in which c.a.i. is available is probably equally well known, although perhaps less commonly discussed.[1]

Since expense is an important consideration only when it is great, I will restrict my discussion to c.a.i. systems whose level of sophistication requires the capabilities and facilities associated with a so-called "large" computer. Generally, this means a convenient and flexible interactive authoring language, input-output control programs to accommodate a number (sometimes a large number) of student terminals, storage facilities for an adequate amount of course material, and above all, a powerful and flexible student response processor. This is not to imply that "small" computers (PDP-8 through the Hewlett-Packard class, say) do not have their place as learning tools- they certainly do, and new instructional uses are constantly appearing. Generally, however, computers of this size are best suited to rather specialized applications in courses at the level we are considering, as they are generally unable to support the functions mentioned above.

THE ROLE OF C.A.I.

If c.a.i. is ever to become more than an expensive showpiece or plaything, then the time, effort, and particularly the expense that the use of this medium entails must be clearly justified in terms of its demonstrable academic benefits. In this article I should like to consider some of the factors that affect the usefulness of c.a.i. as a learning tool in courses presented at the university - or advanced secondary school level, giving special attention to several points that I feel often tend to be overlooked.

The power of c.a.i. as a learning tool manifests itself in its interactive nature, its adaptive and branching capabilities, and its record-keeping ability (to enable the course author to refine and improve the instructional material). These capabilities serve to define the uses to which c.a.i. can be put in any given course. Thus the fact that computers cannot at present engage in extensive and meaningful free dialogue limits the value of c.a.i. in teaching history or philosophy (although the use of the computer as an information retrieval device should certainly not be discounted here).

Just as intensive and meaningful dialogue demands direct personal contact between two or more people, there are other operations associated with the learning process that are better carried out by some means other than c.a.i. The transmittal of information is a good example; books and mimeographed sheets are still the cheapest and most convenient way of conveying information of the kind that is required by all the students of a class. There is little to be said for using a c.a.i. system as a "page turner"- students are rapidly bored by long statements presented on typewriter terminals at the normal rate of 10 characters per second. Perhaps the situation would not be so serious with a video display that could be generated at a faster rate, but even then, the student would probably want to have a hard copy of the printed matter for future reference.

There are other ways of transmitting information that are all too often neglected, particularly when a glamorous technology like c.a.i. becomes available. It frequently happens that the spoken word is the most efficient means of getting a point across, especially when the commentary is directed at the explanation or clarification of a concept that is simultaneously being presented by some other medium. This other medium is usually visual material- either printed or projected (or both), consisting of symbols or complete diagrams. If these diagrams can be assembled or made to evolve by means of animation techniques, then an entire new dimension is available; in some cases the audio and visual media can be combined with c.a.i. at the terminal location, and presentation can be under computer control. For other purposes, it is frequently more practical to present the audio-visual material off-line. Whichever arrangement is followed, it is important that the teacher or course designer be fully aware of the advantages and limitations of all these techniques. It will usually be found that when the various parts of the instructional load are assigned to the different media in a comprehensive way, then the overall learning system is improved far beyond what these media can offer when used in isolation. While this may sound very elementary, it seems to be almost universally neglected in practice where c.a.i. is involved- largely, I suppose, because the people who develop c.a.i. systems and programs are often primarily interested in c.a.i. itself, and their very specialization tends to shield them from this broader view.

Most of the teachers who send their students down the hall to "use" c.a.i. (or audiotapes) do little better, however. So many instructors do not realize that these techniques, when sufficiently well developed, are powerful tools that exert a perturbation on the more conventional aspects of the course. As Neil Postman put it in his engaging CBC Radio talk last year,[2] we need to apply a "philosophy of subtraction"- to recognize that the entire course structure and what goes on in the classroom must be re-evaluated every time a new medium is added. Simply tacking another learning tool onto an existing and unchanging regime of lectures, tutorials, reading, and assignments does little to improve the student's learning efficiency in the course, although the novelty may be welcome at first. If c.a.i. is treated as merely another teaching "aid" without being integrated into the course in a comprehensive manner, it seems unlikely that it will yield benefits enough to justify its cost.

Given even an optimum mix of instructional media and a teacher who is sensitive to the impact of these media on the teaching of his course, one is still faced with the problem of interfacing what is primarily a self-paced individual study method of learning with the traditional two-or-three - lectures-per-week lockstep format of the usual university course. To the extent that these individual study methods are successful, the amount of time devoted to formal lectures can be reduced- but here we come up against "the system": what happens to course credit ratings, teaching load calculations, and our all-important means of justifying faculty appointments? It is perhaps just as well that the development of an individual-study course format takes so long that these changes can generally be introduced piecemeal without unduly upsetting deans and administrators!

I should perhaps remark that I do not believe that the availability of c.a.i. and other media mean that the lecture is dead, even in courses that are largely of the "training" type such as transistor electronics or engineering graphics.

It is hard to conceive of any subject that does not have its historical and humanistic elements, its relations to other disciplines, and its patterns of development that can best be communicated by a living, thinking practicioner of that subject. The primary function of the university teacher is to communicate these ideas, together with his own enthusiasm for the subject, and to place what has already been learned in a proper perspective. The learning itself, however, takes place in the student's own frame of reference (and time scale), which is rarely that of the lecturer. Far from being the death of the lecture, the sensitive and judicious use of c.a.i. and similar learning tools can in many cases revitalize the classroom experience by "liberating" it from those activities that it could never accomplish very effectively.

TEACHERS, AUTHORS AND PROGRAMMERS

C.a.i. is, at its best, an extension of the teacher; a means of enabling the teacher's experience and sensitivity to influence the student in his learning. Because the student-teacher relation is so much more intimate and direct than in the usual classroom situation, it is important that the very best teachers become involved in writing c.a.i. materials. Unfortunately, very few experienced teachers are willing or able to devote their time to these ends. The tradition that one's "teaching" committment extends little beyond presenting three lectures per week in a given course, and the feeling by many faculty that students of university calibre should not really *need* instructional aids (beyond the textbook), do little to encourage these activities. And the peculiar value system of academe that says (at best) "publish and thrive", offers little reward for the time needed to implement and maintain a c.a.i.-based course; one doesn't even end up with a textbook that can be claimed as a "publication" or protected by a meaningful copyright.

With so few teachers directly involved in c.a.i., there is a tendency to delegate much of the writing to others-graduate students, computer personnel, instructional programmers, etc. These attempts almost invariably fail to produce c.a.i. of satisfactory quality, for much the same reason that none of these people would ordinarily be equipped to write a good textbook or deliver a series of lectures in the subject: they have simply not had enough experience to be aware of the multiplicity of difficulties that students experience in learning the subject, or of the subtle nuances of approach that can get the student over these barriers and encourage him to face more.

Since we cannot expect a large number of teachers to involve themselves very extensively in program authoring, it is important that they design their material in such a way that a maximum number of other teachers can make use of it in their own teaching. This often means arranging the c.a.i. program in segments or modules so that the material can be used in different sequences by different classes. It is also possible with many c.a.i. systems to arrange, for different groups of students to proceed through the course material in different trajectories or to be treated in some other special manner.

Since c.a.i. developed from other uses of the computer, it is not surprising that the c.a.i. services at many institutions are operated by the local Computing Centre. Less commonly, Departments or Faculties of Education, where these exist, are responsible for c.a.i., particularly where research or development work on c.a.i. is the principal concern. A third arrangement is to make c.a.i. an entirely academic affair, divorced from any single department of administrative agency of the University. This is the organizational structure we have adopted at S.F.U.; the C.A.I. Centre is directly under the Academic Vice President, whose office provides the necessary funding. The role of the Computing Centre is to furnish the physical equipment and support services, which they do under a "facilities management agreement" with the C.A.I. Centre. By placing the responsibility for the development of c.a.i. in the academic arm of the University, we hope to integrate it more closely into the academic mainstream.

SYSTEMS AND SOFTWARE

While comparatively primitive c.a.i. systems can adequately serve many teaching functions at the elementary school drill-and-practice level, considerable sophistication is demanded if the system is to make a useful and economical contribution to secondary school and college teaching. This sophistication must be reflected in the design of the programming language if the system is to be effectively utilized by teachers.

It is unfortunate that none of the presently available c.a.i. authoring languages reflect the state-of-the-art in computing languages as it exists today. I believe that this is due, in large measure, to a simple lack of communication between language designers (who can hardly be familiar with the problems of teaching real subjects to live students), and the teachers, whose general lack of familiarity with computing science leaves them unaware of what they could reasonably ask for.

All of the principal c.a.i. authoring languages are, in my view, lacking in character string processing capabilities that are essential for adequate analysis of student responses. Others are deficient in arithmetic computing facilities. Very few languages are generalized enough to support a variety of terminal types (both typewriter and video display), nor do they commonly allow for the programming of additional functions and capabilities at other than the assembly (or FORTRAN) level.

The effect of this is to seriously limit the applicability of c.a.i. to the kinds of tasks that it must perform at the advanced high school or college level. Authors are forced to fit their instructional strategies to the Procrustean bed of an inadequate language, and thereby degrade the entire functional capability of the c.a.i. process. Extreme examples of this are seen where non-c.a.i. languages such as APL or BASIC are employed for c.a.i. purposes. What makes this situation so maddening is that many of the things we want to do can easily be done by a combination of languages that presently exist - the problem is mainly one of merging these capabilities into one single language. Some individual centers have begun to do this; Western Washington State College, for example, has merged IBM's Coursewriter III (which lacks decimal- or floating point variables)[3] with their PL/1 - derived on-line conversational computing language, thus adding both arithmetic calculating and string processing capabilities to Coursewriter.

But this brings up another difficulty- that of incompatability between program materials from different institutions. Very few institutions will be able to command the interested faculty and the resources required to develop, on their own, all the c.a.i. programming they could profitably use; the time and effort required to write, debug, evaluate, refine, and maintain a complex c.a.i. program is so great that it would be a waste of resources to expect each institution to do this for every subject anyway. There must be exchange of materials between organizations, and this requires a certain amount of uniformity. Not total uniformity, which would undoubtedly impede progress in the state of the art, so not a single, universal language- but not a hundred different languages, either, and the latter situation is a more accurate description of what we are up against today. There is an unfortunate tendency for individual university computing groups to write their own c.a.i. languages- so often it seems, without any reference to (or even awareness of) the academic functions the language must serve, or the advantages and disadvantages of other c.a.i. languages. What usually results in just another undistinguished c.a.i. language, often very limited in scope, which serves only to isolate the institution from the opportunity to cooperate and exchange materials with other organizations. (This is very much the situation in Canada, and it seems to be getting worse). Faculty are much less interested in devoting large amounts of time to developing programming that has little possibility of use elsewhere, so the c.a.i. bill of fare becomes overladen with the usual clutter of nonsubjects such as "Introduction to FORTRAN" and "How to use the Library".

The requirements of c.a.i. authoring languages are discussed elsewhere in this volume by Karl Zinn, to I will mention only three general points here. First, there is nothing that puts a student off more than to have the computer fail to recognize a response that the student believes to be correct. This means that the programming language should be capable of recognizing keywords or fragments embedded in other material (in the case of a typewriter keyboard input from the student), and there should be some means of making the system ignore the presence or absence of capital letters, punctuation, etc., where these are irrelevant to the response. The inability to do these things conveniently is a common failing of some of the less sophisticated c.a.i. languages, and of APL and BASIC.

Secondly, the author must have a convenient means of detecting the points in his program at which he has failed to provide adequate wrong-answer recognition. This requires, of course, an adequate means of storing unrecognized answers (or ideally, answers of any desired category) for future incorporation into the program. We find this an absolutely essential part of developing a c.a.i. program to the point that we can have confidence in it. For the author, this means that his responsibility extends far beyond the mere writing of the program; refinement and maintenance must continue for several years, and this can be almost as time consuming as writing the original program itself.

Finally, there are many instances where it is desirable to employ audiotape or visual displays in conjunction with c.a.i. Most languages do not provide for the support of such devices directly, or else, as with Coursewriter III, the auxiliary equipment is too costly to be practical.

We have found that a less elegant, but thoroughly useable slide projector control can be effected by indirect means through an ordinary terminal. For example, the typeball of an IBM Selectric terminal can be used to operate microswitches (at the far right-hand margin, beyond the usual typing area) that control forward- and reverse movements of an ordinary projector. Similar control from a videotube display can be had by mounting miniature photocells on the face of the display tube. While audiotape devices can be controlled in the same way, we prefer to use the audiotape (or tone signals recorded thereon) to control the pacing of the c.a.i. program in such a way that the student alternates between listening segments (presented with slides) and brief c.a.i. exercises that reinforce each of these segments.

One new display device that appears likely to become quite popular (and may even displace TV and conventional ciné for all but large group instruction) is the Philips "P.I.P." system, which consists of a Super-8 film transport that is activated by tone signals on an audiotape. Both the audiotape and the film fit into convenient cassettes, and film advance can be as rapid as 24 frames per second, permitting complex animation displays in full color. Since the film is moving only when motion must be depicted, the system is very economical. We are presently investigating the possibility of interfacing this device with a conventional c.a.i. system, and believe that it can provide an inexpensive means of presenting a display of a kind that will not be economically feasible on a direct computer output device for many years to come.

REFERENCES

1.) Oettinger, A.G. "Run Computer, Run: The Mythology of Educational Innovation". Harvard University Press, 1969.

2.) Postman, Neil "Telling it like it ain't": CBC "Ideas" series on Perception and Prejudice, February 16th, 1971. A recording of this talk is available from the CBC for $12, or from me for free (please send a blank 1200' tape).

3.) Lower, S.K., and Arsenault, G. "A Functional Evaluation of Coursewriter III", copies available from the author.

CAL TERMINALS - SOME RECENT CANADIAN INNOVATIONS

Peter M. Carey, John Tate
Lektromedia

CAL TERMINALS - SOME RECENT CANADIAN INNOVATIONS

SESSION 2A
ROOM/SALLE 143

By/Par: Peter M. Carey, John Tate
Lektromedia.

Inadequate terminal facilities has been a major impediment to the development of effective communication between students and course materials and hence to any widespread acceptance of CAL as an instructional aid.

This paper will describe some recent advances made in Canada in terminal design and will be illustrated by operating multi-media terminals of Canadian design representing the latest state-of-the-art.

The areas of economic graphics, random-access audio/visual and touch sense interactions will be covered. The paper will relate a terminal's inter-active communications facilities to the possible instructional strategies, and to the overall costs involved.

Special reference will be made to considerations for authoring and editing multi-media CAL materials.

QUELQUES INNOVATIONS CANADIENNES RÉCENTES DANS LES TERMINAUX DES ORDINATEURS PÉDAGOGIQUES

Des terminaux inappropriés sont à la base des obstacles majeurs au développement de relations efficaces entre les étudiants et les ordinateurs pédagogiques et, de ce fait, à l'acceptation répandue de l'instruction à l'aide d'ordinateurs.

Dans cette communication, nous décrirons de récents progrès, au Canada, dans la conception des terminaux et nous illustrerons ces progrès en faisant fonctionner des terminaux à affichages multiples de conception canadienne représentant la technique la plus avancée.

Nous examinerons les domaines des graphiques économiques et des interactions à volonté et aléatoires de l'audio-visuel. Nous établirons un rapport entre, d'une part un terminal conversationnel et, d'autre part, les méthodes pédagogiques et les prix de revient.

Nous parlerons spécialement de la constitution et de l'édition des cours CAL.

TITLE : COMPUTER AIDED LEARNING - STUDENT TERMINALS - SOME RECENT CANADIAN INNOVATIONS.

By : Peter M. Carey
John Tate Lektromedia, Pointe Claire, P.Q., Canada

SUMMARY : This paper describes some recent advances in computer terminals for CAL and other high-level inter-active uses. The work stems from our contention that the lack of appropriate terminals is impeding the advancement of CAL. Recent developments include CRT terminals with graphics, special fonts, and audio-visual media attachments.

CREDITS : The work described herein was done by Lektromedia under an Industry Research Assistance grant from National Research Council.

Of the various technologies to be applied in Education, Computer Aided Learning promises to have the most profound impact, -even though it is still in its infancy. We do not intend here to discuss the merits, of CAL as a teaching aid, that's discussion material for teachers. We, at Lektromedia, are technologiests and it was our premise in 1970 that the computer-terminals then available were inadequate for CAL. We felt that it was the job of technologists to invent terminals specially for CAL and that until this was done, then CAL would not be taken very seriously by teachers or students.

Up to 1970, most of the terminals used for CAI were teletypes. These are not only very noisy but also force a very low-level of instructional dialog since all interaction must use the printed word, usually at an irritatingly slow speed. Some people were using CRT terminals but even these were data- processing oriented and still used only the printed word like teletypes although they were quiet and faster and, needless to say, more expensive.

One of the first questions we asked was, - "If you have a CRT, isn't some material better presented in graphical or pictorial form ? People get quickly bored if everything is presented in text form". -Witness text-books without any illustrations ! Another idea we pondered, -with due respect to efficiency experts) - was "Don't write it, -say it" --- and so our "talking terminal" came into being.

In response to the first question, I will summarize our approach to squeezing "the most for the least" out of a CRT. You can get an inadequate Text-only CRT for about $ 2,000 and an over-elaborate graphics CRT for $ 20,000. What we wanted was a better compromise between performance and costs, i.e. - a lot more performance for a little extra cost (than the cheaper units).

So, we invented what came to be nicknamed "Lektrographics". This is a graphics generation technique using a 6-bit coded font which can be stored and transmitted the same as ordinary text. We reasoned this way; "why pay $ 20,000 for precision graphics when for a few hundred dollars you could have something nearly as good which is suitable for most instructional uses ?" Lektrographics is just that, - it is an "electronic blackboard" that can be used to illustrate concepts effectively although not to any great degree of precision.

Lektrographics (patents pending) is based on drawing a diagram by joining up adjacent characters using a 6-point and 4-bar font. The dots and bars fill the location normally occupied by a character with its surrounding spaces so, adjacent graphical characters can adjoin to make continuous lines. Curves are approximated by plotting the best approximation of points.

Using this technique, a graphic diagram can be combined with text in CAL sequences and the "level of communication" to the student is noticeably enhanced for a very modest cost.

We also felt it was important to make the graphics easy for teachers to prepare in the authoring and editing stages. - If otherwise, then it wouldn't be used. Terminals are physically arranged so that any unit can present graphics in the student mode. To author graphics, the teacher plugs in a small mimic-style composing board. This not only prevents students from modifying a graphic but also optimises the price because only one composing board is needed among several terminals.

The next question we addressed was the need for terminals to present material using familiar symbols so that they are easily comprehensible. For example, most terminals present text written only in upper case which is unnatural; nor can they produce conventional mathematical and scientific symbols. These limitations detract from the ability of the CAL to teach.

So we designed a programmable font. Like the graphics, this provides full-font characters but on interchangeable modules with characters in groups of 16. You can have, for example, a bilingual upper/lower case set with French accents such as ç, é, etc. You can have two full alphabets for language instruction, say French-Russian. You can have sub-scripted and exponential characters. You can have mathematical and scientific symbols such as Σ , $\int$, $\sqrt{}$, x^2 , x_n, π , ∞ , etc. or special symbols such as APL commands. These character sets can be interchanged so that a terminal assigned to, say, mathematics instruction, can be converted to, say, physics or chemistry simply by putting in the modules appropriate to the course content.

The next problem centered on the presentation of visuals because the instructional value of CAL curricula can be greatly enhanced by using slides, film or T.V. These must however be randomly accessible if the full adaptive power of the computer is to be of value.

So we designed the terminals to accept and display T.V. images and also to synchronise into a T.V. system so that T.V. pictures can be combined with terminal data.

We then designed a random-access slide selection system. This at present accomodates up to 100 - 35 mm slides but can later be expanded to include strip-film, - still frames or movie. The terminal displays the optical image immediately adjacent to the computer-generated (CRT) image giving the overall appearance of a single screen with side-by-side images.

This approach was taken partly to enable the student to view both images together (since usually they are related) and also so that the full screen can be made responsive to touch. For the student to always enter his responses and questions through a keyboard is very restricting, especially to small children who often can't spell. The touch-sensitive display allows a very natural and effective dialog since the student can touch his finger anywhere on the slide or CRT image. The computer can discriminate the position of his finger to about 1/10", -accurate enough for most purposes.

The touch-sensitive device was originally invented at NRC but is still somewhat expensive. Despite this, we have shown that it provides a powerful input device and either it, or competing devices, will soon be feasible for general use.

Perhaps our most significant innovation is the "talking terminal". This is a casette-based random-access audio system built into a remote terminal. Each casette holds up to 100 individual messages of 60 minutes total playing time. Maximum access-time is 60 seconds, end-to-end, although if messages are organised effectively, an access-time of 1 - 5 seconds would be more normal. The audio quality is excellent, in fact, it is almost hi-fi reproduction. The audio terminal is "smart" in that it contains a microprocessor. This is a "must" in handling the audio system but also serves as an effective means of header-word processing, slide and touch panel control and implements certain CRT editing and formatting functions.

The micro processor is programmed to operate the audio system in either student or author mode. Physically all audio terminals operate in the student mode. To do authoring, you plug in an extension authoring board into any student terminal.

In authoring, the terminal is placed off line and can be used to generate new tapes or edit existing tapes. During authoring of a new tape, an index containing the start and stop address of each record is generated and resides on the tape itself. During the editing phase this index is updated continuously. By means of the index, all call-up of messages by the CAI programs is by means of message number (00 - 99).

The student mode starts with an initialization whereby the index is automatically unloaded from tape into the terminal memory. At this time, the identity of the tape is checked against that required by the CAI. When the CAI program subsequently calls up a particular audio message, the index is used to determine the location of that message on the tape, search the tape at high speed, play the message and stop at the end of it.

At Lektromedia, we believe that in order to be cost effective in future, it is desirable to adopt a system approach of concentrating terminal control into central units which may drive a cluster of local terminals. This will reduce hardware costs and more significantly, allow remote operation of the whole cluster over a single high speed communication link.

The designs we have described have been packaged into two basic models of terminals, known as LEK 110 and 120. The first includes only the various CRT features. In this case, the programmable fonts and graphics are accessed via Shift-Out / Shift-In control characters. The second model, LEK 120, has all the same CRT features but also includes audio/visual and touch screen facilities which are accessed by means of a header-word system. All models can be operated remotely from the computer over a telephone line.

All the development work of these terminals has been done in Canada in consultation with NRC with particular emphasis on technical standards. This is to ensure that any terminal will operate with any computer and that curriculum software can be inter-changed among users. In this regard, the communication coding is important, particularly in regard to the coding of functional controls and the French accents. A uniform header-word format for remote multi-device terminals has also been followed.

It will be evident that these terminals enable the communication between student and CAI material to be at a very high level, -compared to previous technology. This permits a much more comprehensive and natural dialog and greater flexibility in selecting among alternate strategies for instruction.

For example, the graphics offer a low-cost but very powerful means of communicating with pictures as well as words. A terminal without graphics is like a teacher without a black board. The programmable font makes mathematic and scientific instruction practical rather than clumsy. The audio and foreign alphabets mean that language instruction by CAL can now be considered seriously. The random-access slides and touch sensitive screen open the door to widespread use of CAL for small children.

It is by using a high-level communication in combination with the full adaptive powers of the computer and sophisticated curricula design that one can make CAL truly "effective".

Now, a word about cost. -How can one answer the common complaint, "It's too expensive". -What should a terminal cost? Where are these "low-cost, do-everything" terminals that everyone hangs their hopes on ? -Let me just say that the terminals described herein are here today and they work. They do indeed cost a bit more than devices which are of no use ! The overall cost of a CAI must be related to its effectivity, i.e., Does it teach ? An overall CAI installation built with effective terminals and programmed to utilise them fully will cost more than a CAI system with crude terminals having significantly less teaching value. Nobody has yet incorporated a truly effective CAL system into the educational system, for enough time, and spanning enough material to realise any real pay-off. I can confidently say that these terminals do allow the high-level of communication necessary for effective CAL curricula and that they are absolutely unique.

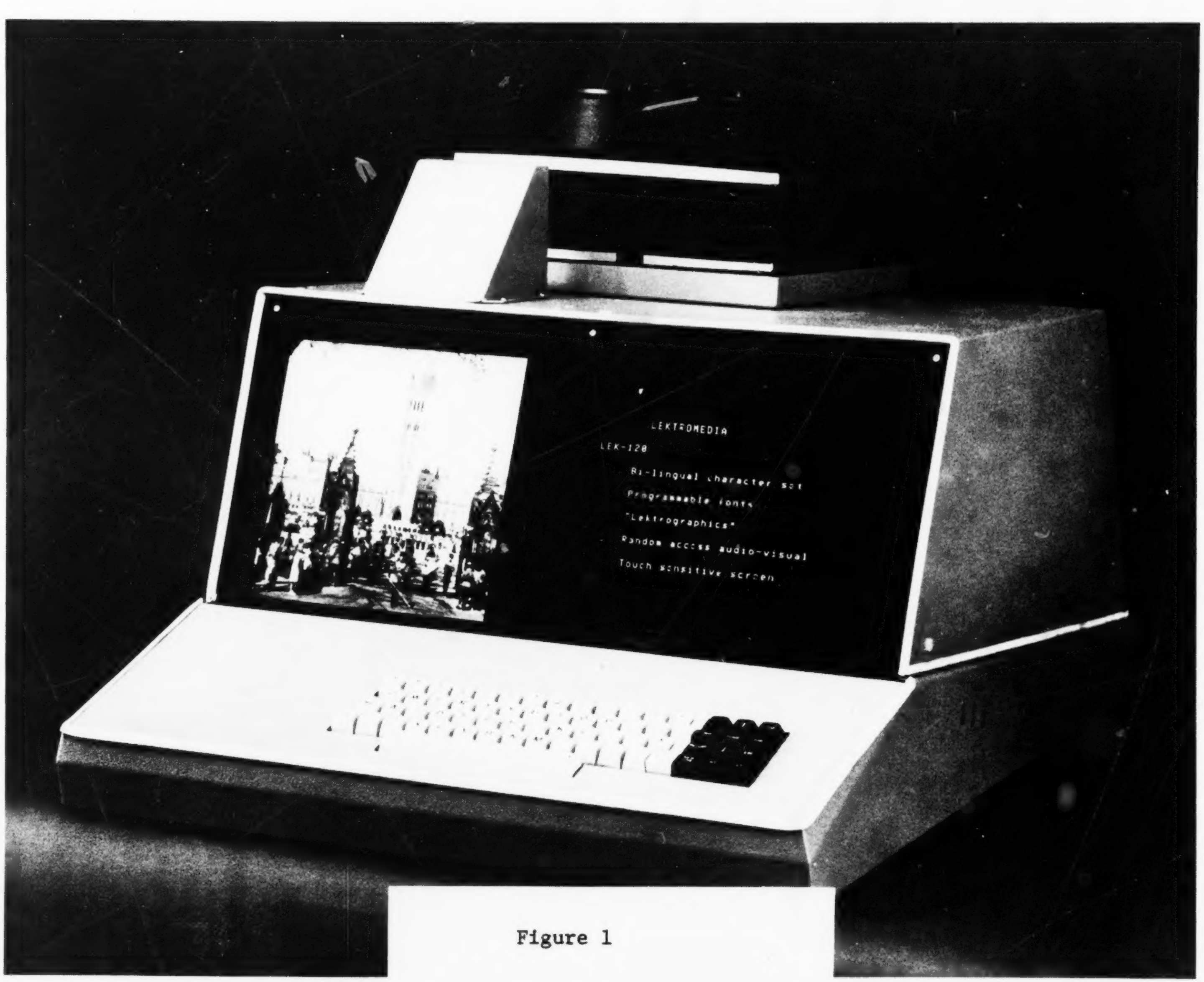

LEKTROMEDIA
LEK-128
Bi-lingual character set
Programmable fonts
"Lektrographics"
Random access audio-visual
Touch sensitive screen

Figure 1

THE ROLE OF THE MINICOMPUTER IN CAI

P.A.V. Thomas, Ph.D.

Electrical Engineering Department

University of Windsor

Windsor, Ontario, Canada

THE ROLE OF THE MINICOMPUTER IN C.A.I.

SESSION 2B
ROOM/SALLE 143

By/Par: P.A.V. Thomas,
University of Windsor

This paper will deal with the concept of using a specially designed graphics system that is being developed about a minicomputer to appraise the capability of carrying out computer aided instruction at a modest cost in remote locations where it may not be possible to provide a large scale system with time shared terminals. At this stage most of the system hardware has been constructed and the software has commenced. The system is constructed about a 12-bit word mini-computer with a 4k-core memory; the graphics facility consists of a logic character generator with a high degree of flexibility, a line generator, a 4k-core refresh memory, 8" x 10" display tube and a light pen. As back up storage for display files, a magnetic tape cassette drive is included in the system. For interactive working there is in addition to the light pen, a teletype and a set of push buttons. By providing hardware generators the amount of data storage is minimised and their novel design is such that they could be easily produced in quantity by L.S.I. techniques thereby keeping the cost to a minimum.

LE RÔLE DU MINI-ORDINATEUR DANS L'EAO

On traite du système graphique spécialement conçu pour un mini-ordinateur afin d'évaluer la possibilité d'effectuer l'instruction à l'aide d'ordinateurs à un prix de revient modeste, dans des endroits éloignés où il n'est peut-être pas possible de fournir un grand système avec des terminaux à temps partagé. Actuellement, la plupart des composantes sont construites et on a commencé à faire la programmation. Le système est construit autour d'un petit ordinateur à mots de 12 bits avec une mémoire en tores de ferrite de 4K mots; la partie graphique se compose d'un générateur de caractères logiques très souple, un générateur de lignes, une mémoire de rappel de 4K mots, un écran de 8 X 10 pouces et un crayon lumineux. On dispose de mémoires secondaires en cassettes pour les dossiers d'images. Pour les conversations avec l'ordinateur, on dispose du crayon lumineux, d'un télétype et d'un groupe de boutons poussoirs. En fournissant des équipements générateurs, on minimise l'emmagasinage des données et, grâce à leur conception originale, on pourrait facilement les produire en quantité par les techniques de microminiaturisation et ainsi réduire au minimum le prix de revient.

ABSTRACT

This paper will deal with the concept of using a specially designed graphics system that is being developed about a minicomputer to appraise the capability of carrying out computer aided instruction at a modest cost in remote locations where it may not be possible to provide a large scale system with time shared terminals. At this stage most of the system hardware has been constructed and the software has commenced. The system is constructed about a 12-bit word minicomputer with a 4k-core memory; the graphics facility consists of a logic character generator with a high degree of flexibility, a line generator, a 4k-core refresh memory, 8" x 10" display tube and a light pen. As back up storage for display files, a magnetic tape cassette drive is included in the system. For interactive working there is in addition to the light pen, a teletype and a set of push buttons. By providing hardware generators the amount of data storage is minimised and their novel design is such that they could be easily produced in quantity by L.S.I. techniques thereby keeping the cost to a minimum.

1. Introduction

Before describing the details of the system developed at the University of Windsor it is desirable to consider some cost factors. In a recent article by Pottle et al [1] a comparison of costs is given for their CDC 3600 system with 60 teletype ports and an estimate of costs for two minicomputer configurations; their figures are summarised in the Table below together with an estimate of the total cost with maintenance. It should be pointed out that these costs were for local systems not involving long distance line costs. The interesting thing to note is that the user costs are comparable for both systems, however, there will obviously be greater "computing power" in one CDC3600 than 60 minicomputers so that one cannot consider the two systems identical, though for many instructional purposes they could be considered as such. If one is only considering the teaching of computer programming one could conceive an even cheaper system costing about $6000 with a user cost of about 60¢/hr. and at the other end of the spectrum there is the PLATO system developed at the University of Illinois costing about $5M

SYSTEM	COST/USER HR.	TOTAL COST	TOTAL COST/ USER
CDC 3600	$5.00	$ 2 - 3M	$25 - 40K
DEC PDP11	$3.30 - $5.00	$30 - 40K	$30 - 40K

TABLE - COST COMPARISON

and claiming projected user costs as low as 50¢/hr. Further similar figures are given in another article by Howard et al [2]. The main point to bring out here is that the user cost/hr can be of the same order irrespective of the size of the system but the initial cost can only be justified if one has a large number of users (or possibly other uses) for the system.

Given a large justifiable capital sum then a large system will probably be more flexible and powerful than a large number of minicomputers, but if the capital sum available is limited, then using one or two minicomputers will give similar user hourly costs. With this background it seemed a worthwhile venture to appraise the CAI capability of a small system costing under $20K for in this way a small establishment far from a large computing system would be able to take advantages of CAI (possibly in a resource centre) at modest cost.

2. Overall System

The overall system shown in Figure 1 was designed to give maximum flexibility in order that different features could be tested out — specifically a CRT with refresh memory is used rather than a storage CRT so that a light pen could be used for interactive purposes. It was decided therefore that the complete system would consist of a PDP-8 minicomputer, a display tube with refresh memory, light pen, keyboard and ultimately a magnetic tape back-up memory. Due to the desire to use a refresh memory and yet be able to display a reasonable amount of data in any one frame, hardware generators were developed to produce the various data forms, specifically, an alphanumeric character generator and vector generator.

3. Data Form

Before considering the details of the system it is necessary to describe the form of the data used in a display. Essentially this can be said to consist of three types: *(i)* data applying to a particular item, be it alphanumeric character or a line segment, and referred to as an Item Data, *(ii)* data applying to a particular mode which energises a particular graphical generator and referred to as a Mode Data, and *(iii)* a Jump Instruction (Jump Data) which permits a particular display file to be refreshed or entry made to a new file. There are a number of methods used to organize such a display file and due to the short word length of the particular computer (12 bits), these three types of data each occupy one word in a display file. This naturally leads to three different types of data format which are shown in an Appendix. It will be observed that the item data always begins with a zero and the "control instructions" with a one. Furthermore the second bit of an "instruction" is a zero for a mode control and a one for a jump. This coding assists both the hardware design and also the user for programming.

4. Refresh Buffer Memory

The primary purpose of this buffer is to match the speeds of the input/output (I/O) transfer of data through the computer (approximately 4000 words/sec) to the graphic item generation (1μs to 40μs). Various forms of buffer storage were considered, that is, drums, discs, delay lines, and core storage with respect to speed and the required capacity which should not be less than 2000 words. The ultimate decision was to use a core memory of 4096 words similar to that used in the computer. This not only gives adequate space but also eased the design.

Associated with this memory there are two single word buffer registers and two addressing registers for loading of data into the memory from the computer and accessing the data to the various graphic generators. Furthermore having accessed the data from the memory it is necessary to have a Controller capable of interrogating the data and determining the required action. The complete system, shown in Figure 1, has been fully described elsewhere [3] and will be only briefly described here.

When initially turned on the system is essentially static. A computer override command will normally be given from the computer through a device selector to the Data Flow Controller (DFC), to give the computer complete control of the memory. Under these circumstances the normal sequence is to first load into the memory I/O Address Register (IOAR) the location of the first data item to be loaded using a Memory Address Load command. Following this an Input Load command is given which transfers data from the computer to the I/O Buffer (IOB) and then triggers a write cycle of the memory to transfer the data from IOB into the memory location specified in IOAR. This process may be repeated as often as required; however, in practice if a file is being loaded all data will occupy sequential locations so that automatic incrementation of IOAR may be initiated using an automatic increment IOAR command thereby saving the time of continually loading IOAR.

When a file is completely loaded and a display is required, the computer override is turned off which automatically starts the refresh side of the memory with the Display Address Register (DAR), and Buffer Register (DBR) cleared. The first read cycle from the memory therefore automatically starts from location 0 and transfers the contents of this location to DBR.

Typically there may be a number of different display files as shown in Figure 2 in the memory at any one time so that the item in this first location (0000) will normally be a jump instruction to the beginning of the desired file. As seen in the Appendix, two types of Jump are available; the reason for this is that the direct jump is limited to 256 memory locations (due to the 12-bit word length) whereas a file may be longer than this and certainly located more than 256 words from the beginning of memory. The indirect jump permits jumping to any location but requires two memory cycles. Furthermore, when using the direct jump a page bit (bit 3) is used to indicate a common zero page or the current page (each page being 400_8 locations). Normally, therefore, the first location will contain an indirect jump, detected by bits 0 - 2 being '1' and the next word is accessed to obtain a pointer address to which the jump is to be made. If on the other hand a direct jump were used the core location portion would be transferred into the DAR, bits 0 - 3 being cleared if the page bit is zero.

The next item to be accessed will generally be a Mode data and will be decoded as such due to bits 0 - 1 being 10. This will turn on the appropriate Generator (A/N, Vector or Dot, to be described later) and set condition flip flops particular to that generator. At the same time the DAR is automatically incremented. The next item of data is then immediately read into DBR and in this case will normally be an Item data which is detected as such by bit 0 being a '0'. The active generator is then checked for readiness to accept data whence the data is transferred to its own buffer register; again DAR is incremented and a memory read initiated. Further Item data are displayed until another Mode data is detected when either the condition flip flops in the active generator may be changed or the active generator is turned off and another turned on. This process continues until another Jump is detected; normally this jump will be a direct jump to location 0 which then refreshes the display file.

If now the file has to be modified, control can be returned to the computer by means of the computer override feature mentioned earlier. Alternatively "cycle stealing" may be used by giving an Input Load Command which loads IOB and signals a Request to DFC. As the refresh side is asynchronous and data generation may take a greater time than a memory cycle (8 μs), further data cannot be supplied to the generator buffer; the I/O Request is now detected and the contents of IOB written into memory location specified by the contents of IOAR.

Having described the methods of memory access by the computer and display tube, it is appropriate at this point to discuss the method of interaction using a light pen. A light pen register (LPR) normally tracks the current value in DAR; if however a light impulse is received by the light pen and the particular item being displayed is light pen sensitive (determined by bit 9 of the last Mode word) a signal is supplied to DFC which generates a program interrupt to the computer and inhibits any further change in LPR, whilst allowing the display to continue. In response to the interrupt the computer can then read the contents of LPR through IOB; this identifies the item "picked" by the light pen, and the user can make any required changes through the computer. After servicing the interrupt the LPR is reactivated so that it continues tracking DAR.

5. Alphanumeric Character Generator

The alphanumeric character generator is a 5 x 7 dot matrix generator that was implemented entirely with NAND logic and described elsewhere [4,5]. Briefly, the matrix is formed using Y- and X-up-down counter registers feeding digital/analog converters (DACs) the former being incremented up to 7 by the main clock and the latter being incremented to 5 on each 7th pulse and a further double count after this to provide spacing between characters; at the same time a brightening pulse train is generated according to the character to be displayed.

Three sizes of character are available, the size being controlled by altering the counting position of the 10-bit X- and Y-counter registers. Basically this permits a display area of 1024 x 1024 raster units; in the case of character generation the smallest size uses increments of 2 raster units/dot position of the matrix but by altering the counting to the 4 and 8 bit positions the character size is doubled or quadrupled permitting 3 character sizes under program control. Furthermore a small character normally considered as a subscript may be moved up one character height for superscripting purposes. Due to this method of generation it should be pointed out that there is a minor limitation in that a character can only be positioned at discrete positions, being a multiple of one character size.

The ASCII character to be displayed is loaded into a Character Buffer, 7 bits in length, whereas only 6 bits are used for the character selection. The reason for the 7th bit is to allow for certain special characters, back space, line feed, character return (corresponding to carriage return on the teletypewriter) and tabulation which have similar 6-bit codes as some of the alphanumeric characters. In these cases all 7 bits are decoded and if a special character is decoded, character generation is inhibited. The operations performed are self explanatory and will be only briefly mentioned. Line feed may be single or double depending on the value of bit 4 in the data word which controls the decrementing of the Y-counter register. In the case of character return the return is again controlled by bit 4, being to the left hand margin (by clearing the X-counter register to zero) or to one of 8 positions equally spaced across the screen depending on the value (0-7) previously set in a special 3-bit Margin Register. This register is set by a Data word (whilst in the alphanumeric mode) having the desired number in its bits 2-4, the remaining bits being zero corresponding to a 'Null' character which is also a special character and detected as such to inhibit any generation. Tabulation may also be performed and automatically places the following character at the next margin position.

6. Dot Generator

It is intended to provide two methods of line drawing, these being a Vector Generator and a Dot Generator, the former being relatively slow compared to the latter, but more flexible in line positioning. The dot generator, as constructed permits a dot to be generated after incrementation and/or decrementation in both the X and Y direction by 0, 1, 2, or 4 raster units, this being achieved by counting at the appropriate point of the X- and Y-up-down counter registers. Furthermore a series of equally spaced dots up to a maximum of 15 in the same direction may be generated by loading a repetition counter, within the generator, with the number specified in bits 2-5 in the data word.

In addition, a vector generator using analog integrators is under construction which will generate lines of any length and at any angle. At this stage only point or positioning is possible in this mode, this being carried out by loading the required absolute values of X and Y into the corresponding counter registers and applying a brightening pulse after the DACs have settled in the case of the point mode.

7. Brightness Control

Three basic levels of brightness will be available, these being set by bits 10-11 of the Mode words; furthermore when the vector generator has been completed a particular line may be blanked out by bit 1 of the X-data word to save unnecessary vector mode words. A further feature is required when in the alphanumeric mode to compensate for the character size. When changing character size the apparent brightness of a character increases as the size is reduced due to the dots being closer together; to compensate for this 3 more levels are required, these being under the control of the size bits of the Mode and Data words. One other factor that effects the brightness of the display is the length of the file which causes a variation of the refresh rate, the rate decreasing as the file lengthens. To overcome this the file length may be kept constant by filling the unused portion with Null characters whilst in the alphanumeric mode. This, however, is inconvenient and wasteful of buffer memory space, so that a frame synchronisation feature is made available in the vector mode word (bit 6); by sensing this bit a picture frame may be started every 1/30 second irrespective of the length of the file, this being the longest practical time to prevent flicker of the display.

The only other feature provided is provision for 'blinking' any Data item by setting bit 1 of the data word. In this case the effective refresh rate for these data items is reduced to 3-3/4 frames/sec which gives a satisfactory flicker.

8. Results

Although the supporting software has not yet been developed, some displays are shown in Figure 3; specifically *(a)* shows the complete available character set and *(b)* the 3 different character sizes; this also shows the ability of superscripting which is particularly useful in algebraic expressions. *(c)* and *(d)* show the hardware tabulation and variable margin return useful for columned data. *(e)* and *(f)* show the use of lines mixed with alphanumeric data as may commonly appear in CAI material.

References

(1) *"On Important Current Issues Concerning Computers in Electrical Engineering Education"*, C. Pottle et al., IEEE Trans. on Education, p.169, vol.E-14, no.4, (Nov. 1971).

(2) *"On-Line Computer Systems for Engineering Education - State of the Art"*, J.A. Howard et al, p.210, ibid.

(3) *"A Data Flow Controller and Refresh Memory for a Computer Display System"*, D.L. Murray, MASc. Thesis (Windsor 1970).

(4) *"An Alphanumeric Text Generator for a Computer Display"*, W.E. Mennie, MASc. Thesis (Windsor 1970).

(5) *"A Logic Character Generator for Use in a Computer Graphics Facility"*, P.A.V. Thomas and W.E. Mennie, Proc.4th Can. DECUS Symposium (1970).

Acknowledgement

The author wishes to thank the National Research Council for their financial assistance to this project.

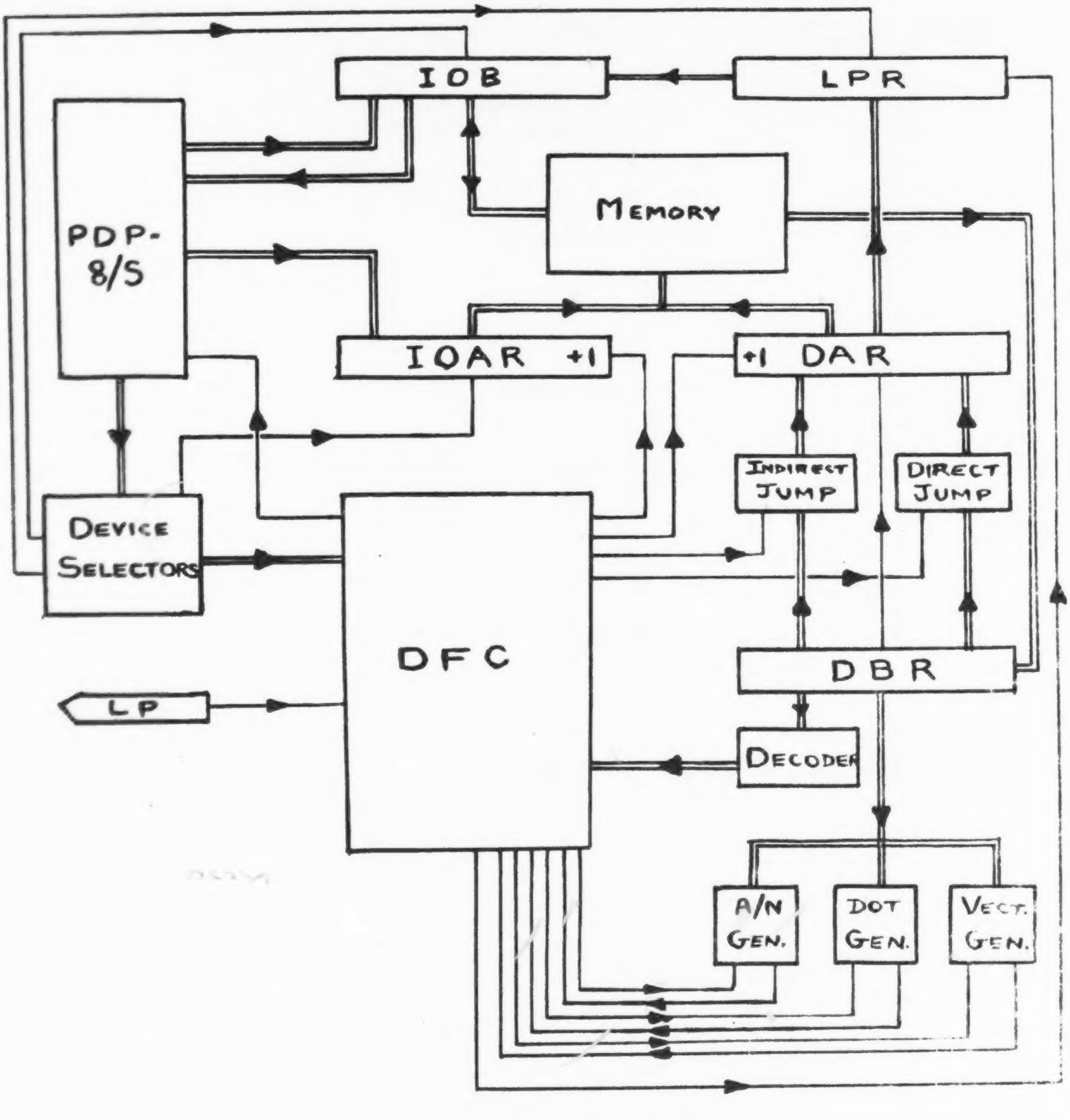
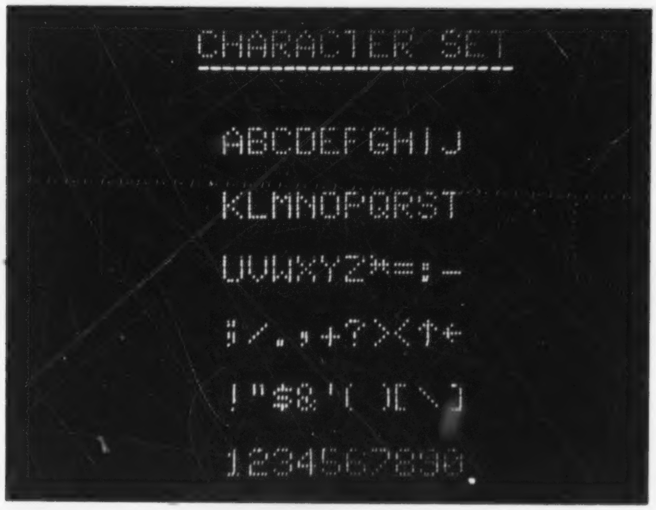

IOB
LPR
PDP-8/S
Memory
IOAR +1
+1 DAR
Indirect Jump
Direct Jump
Device Selectors
DFC
LP
DBR
Decoder
A/N Gen.
Dot Gen.
Vect. Gen.

FIGURE 1 OVERALL SYSTEM

MEMORY LOCATION	CONTENTS	
0000	JMP I	
0001	PTR to absolute address of file	
0002	VECTOR MODE (position only)	FILE I
0003	X	
0004	Y	
0005	A/N MODE	
0006	A/N Code	
⋮	⋮	
0112	JMP 0000	
0113	VECTOR MODE	FILE II
0114	X	
0115	Y	
0116	A/N MODE	
0117	A/N Code	
⋮	⋮	
0400	⋮	
⋮	⋮	
0421	JMP 0000	

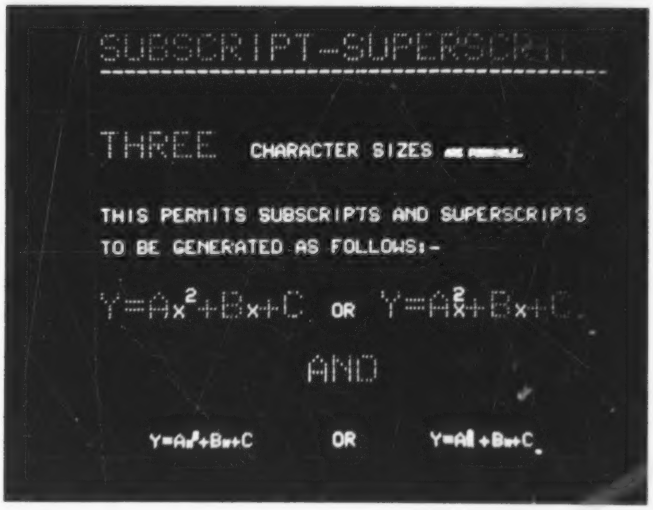
Figure 2. Typical Display File



```
CHARACTER SET
-------------

ABCDEFGHIJ
KLMNOPQRST
UVWXYZ*=;-
#/.,+?><↑←
!"$&'()[\]
1234567890.
```

(a)



SUBSCRIPT-SUPERSCRIPT

THREE CHARACTER SIZES

THIS PERMITS SUBSCRIPTS AND SUPERSCRIPTS
TO BE GENERATED AS FOLLOWS:-

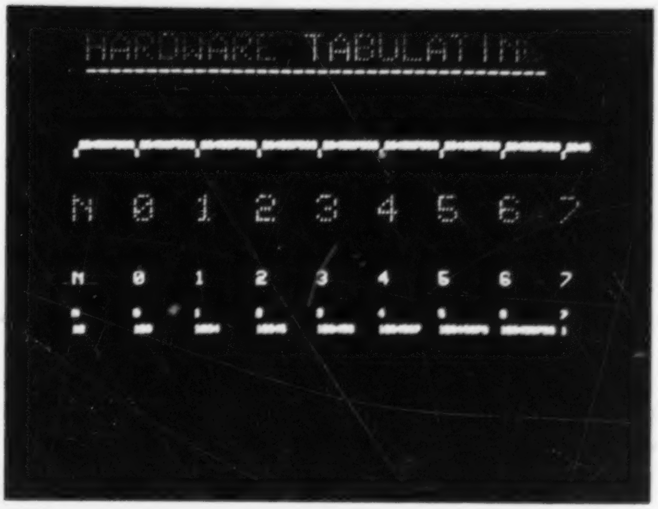
$Y=A_x^2+B_x+C$ OR $Y=A\overset{2}{x}+B_x+C$

AND

$Y=A_x^2+B_x+C$ OR $Y=A\overset{2}{x}+B_x+C$.

(b)

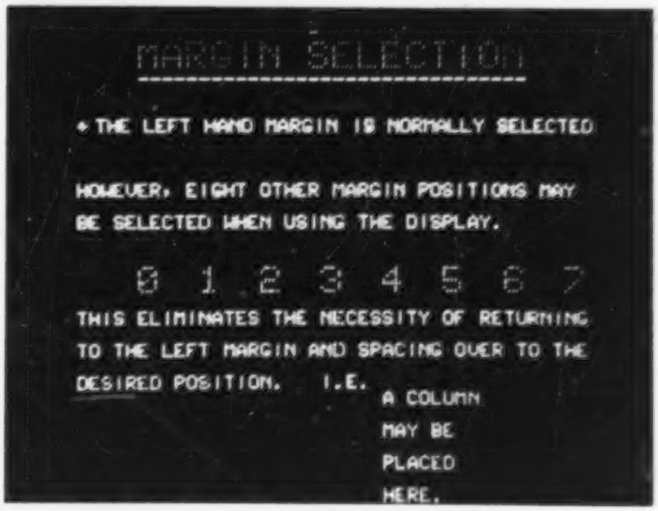
FIGURE 3


HARDWARE TABULATING

N 0 1 2 3 4 5 6 7

N 0 1 2 3 4 5 6 7

(c)


MARGIN SELECTION

* THE LEFT HAND MARGIN IS NORMALLY SELECTED

HOWEVER, EIGHT OTHER MARGIN POSITIONS MAY BE SELECTED WHEN USING THE DISPLAY.

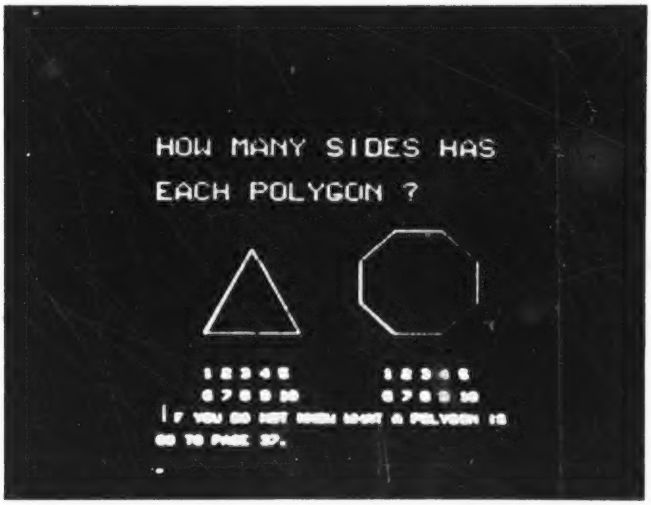
0 1 2 3 4 5 6 7

THIS ELIMINATES THE NECESSITY OF RETURNING TO THE LEFT MARGIN AND SPACING OVER TO THE DESIRED POSITION. I.E.

A COLUMN
MAY BE
PLACED
HERE.

(d)

FIGURE 3

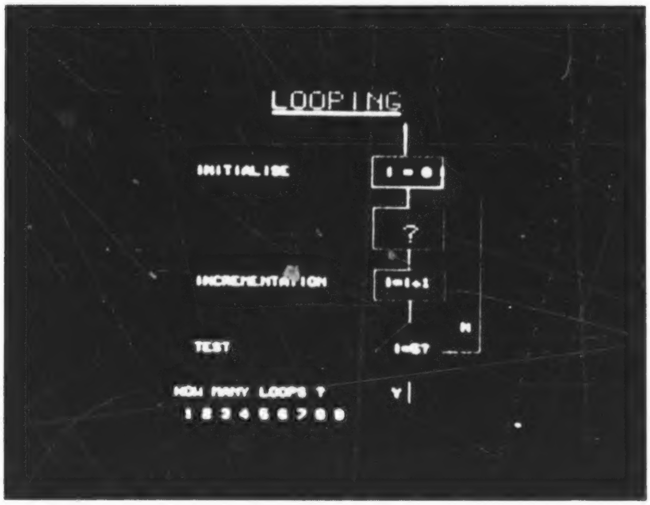


HOW MANY SIDES HAS EACH POLYGON ?

1 2 3 4 5
6 7 8 9 10
1 2 3 4 5
6 7 8 9 10

IF YOU DO NOT KNOW WHAT A POLYGON IS GO TO PAGE 37.

(e)



LOOPING
INITIALISE
I = 0
?
INCREMENTATION
I=I+1
TEST
I=6?
N
Y
1 2 3 4 5 6 7 8 9

HOW MANY LOOPS ?

(f)

FIGURE 3

APPENDIX

Data Forms

MODE DATA

Vector Mode. Bit positions

0 1 2 3 4 5 6 7 8 9 10 11

1	0 0 0 0		F	T	L	I

Dot Mode.

1	0 0 0 1		L	I

Alphanumeric Mode.

1	0 0 1 0	M		S	L	I

JUMP DATA

Direct Jump.

1	1 0	P	Core Location

Indirect Jump.

1	1 1	

ITEM DATA

Vector Mode. 2 words

0	Z	X co-ordinate
0	B	Y co-ordinate

Dot Mode.

0	B	R	Δ X	Δ Y

Alphanumeric Mode.

0	B	SS	M	7-bit ASCII

Margin Load.

0		MS	0 0 0 0 0 0 0

Symbols used

F : 0 - No frame synch.
1 - Frame synch. on
T : 00 - Position only
01 - Dot at tip
10 - Dashed line
11 - Solid line
L : 0 - Light pen insensitive
1 - Light pen sensitive
I : 00 - Normal
01 - Dim
10 - Bright
M : 0 - Normal
1 - Load next word into Margin register
S : 0 - Normal size
1 - Double size
P : 0 - Page Zero
1 - Current page
Z : 0 - Normal
1 - Blank (overrides I)
B : 0 - Normal
1 - Blink
R : 1 - 15 - No. of repeats
SS : 00 - Normal
01 - Superscript
10 - Subscript
11 - Capital
M(CR) : 0 - Normal
1 - Column return
M(LF) : 0 -Double space
1 - Single space
MS : 0 - 1st.position
to
7 - 8th.position

A MODEL OF A CENTRALIZED CAI SYSTEM

Robert S. McLean

The Ontario Institute for Studies in Education

A MODEL OF A CENTRALIZED CAI SYSTEM

SESSION 2C
ROOM/SALLE 143

By/Par: Robert S. McLean
The Ontario Institute for
Studies in Education

This paper is a preliminary report of an anlaysis of a computer-assisted instruction system consisting of students located at a location remote from the central computer facility and communicating with it through a common data communications channel. The analysis is based on a computer model of the system developed by O.I.S.E. and N.R.C. The model examines the requirements of the data communications channel to provide for the number of terminals to be supported and response times to be maintained. The model is described and additional observational data is cited to support the model's assumptions about characteristics of students' terminal behaviors.

MODÈLE D'UN SYSTÈME CENTRALISÉ D'ENSEIGNEMENT À L'AIDE DE L'ORDINATEUR

Cette communication est en soi un rapport préliminaire sur un système d'instruction à l'aide d'un ordinateur et comportant des étudiants loin de l'ordinateur mais reliés à lui par une voie commune d'échange de données. L'analyse du système est basé sur un modèle de l'ensemble mis au point par l'IORE et le CNRC. Le modèle implique l'étude des conditions imposées de liaison pour que l'on dispose du nombre de terminaux nécessaire et du temps de réponse à être conservé. Nous décrivons le modèle et nous citons des données supplémentaires d'observation pour étayer les hypothèses faites pour établir le modèle et concernant le comportement des étudiants aux terminaux.

The National Research Council (NRC) and The Ontario Institute for Studies in Education (OISE) have developed jointly a computer-assisted instruction (CAI) system for research purposes. NRC provides the central computing facilities for carrying out CAI research while OISE provides the curriculum to be used on that facility. OISE is responsible for the data communications for the project. This paper examines the data communications for that system.

The hardware system is shown schematically in figure 1. Student terminals, typically of the model 33 teletype variety, are connected to a small computer located at OISE. This computer performs a line concentration function enabling all of the student terminal information to be transmitted over one data communication line to the NRC computer in Ottawa. At NRC, a general purpose time-sharing computer executes the CAI program. Output to the student is then returned along the data communications line to the line concentrator which directs it to the appropriate student terminal.

The central computer facility is a general purpose time-sharing computer which is used by other educational institutions for research in instruction. Thus the other lines running into the NRC computer represent additional computing requirements simultaneously with the use from OISE. It should be noted that some of these other users are also using the CAI system developed by OISE.

The CAI system is built around an interpreter for the CAN (Churchill, et al, 1970) CAI language. The interpreter currently is written in FORTRAN and contains additional assembly language communication routines to deal with the line concentrator.

The CAN interpreter is run as an individual job on the NRC computer. It provides the CAI function for as many terminals as can be concentrated onto the data line. Alternatively, it can be used at a single terminal without a concentrator by any time-sharing user. Several copies of the interpreter may be in use at the same time.

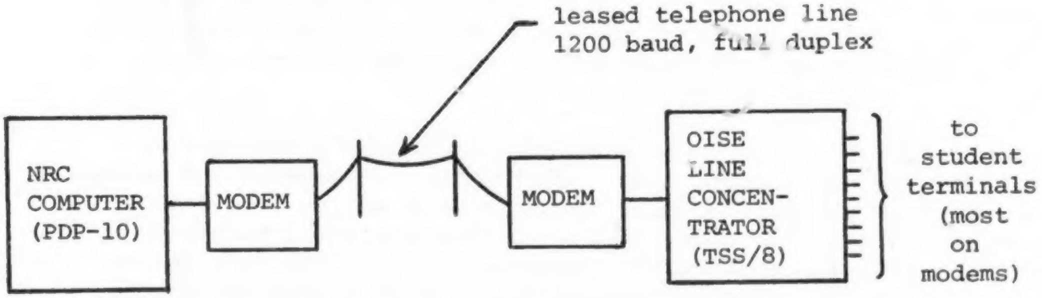


Figure 1. NRC--OISE CAI System

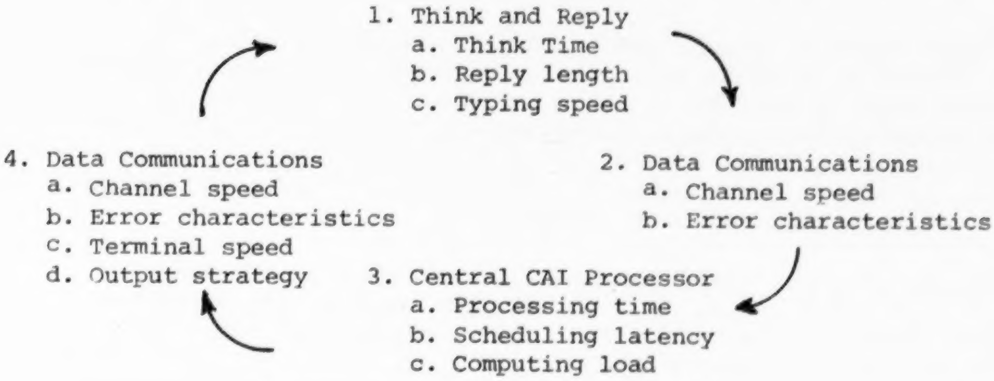

1. Think and Reply
a. Think Time
b. Reply length
c. Typing speed
2. Data Communications
a. Channel speed
b. Error characteristics
3. Central CAI Processor
a. Processing time
b. Scheduling latency
c. Computing load
4. Data Communications
a. Channel speed
b. Error characteristics
c. Terminal speed
d. Output strategy

Figure 2. The CAI message cycle

The CAN system is now operational. It provides a research facility for the testing and evaluation of CAI curricula in remedial mathematics for up to 10 students simultaneously. The design choices that have been made for this system represent one of many possibilities, however, and it was considered that a model of this type of processor organization might indicate the feasibility of similar designs using a variety of communications equipment.

The Model's Structure

The model is composed of four major components. These may be viewed as a circle of events as indicated in figure 2. The model maintains the steady-state performance of N of these cycles representing the N students who are using the system at a given time. Hence, the number of students using the system is one of the major parameters determining the behavior of the system. Additional parameters which govern system behavior are indicated in figure 2 opposite the portion of the model that they influence.

The cycle is initiated following the termination of output to the student from step four. Student behavior consists of 1) an amount of time required to think about the output from the computer and to construct a reply, 2) the length of the reply that the student constructs, and 3) the typing speed with which he can enter the response on his terminal.

The second phase in the cycle begins with the student's response as entered into the line concentrator. Identifying information and message protocol information must be added to the message and this is denoted as message overhead. The speed with which messages are transmitted to the central computer and the error characteristics of the data channel determine the time required to transmit the student's response to the computer. Message queuing within the line concentrator is on a first in, first out basis under all conditions.

Once the message has arrived at the central processor, four parameters govern the length of time of the interaction. These are the availability of the central processor, the length of processing time to produce a response, the length of that response, and the amount of time that a given job can spend executing within the time-sharing computer. The scheduling latency and slice time parameters are functions of the time-sharing monitor's queue discipline and of the amount of computing activity at any given instance.

The response length is primarily a function of the curriculum itself. The length of processing required for a given interaction with a student is a function both of the curriculum and the interpreter that executes that curriculum.

The transmission to the student is primarily a function of the channel speed between the central computer and the line concentrator and the speed of the student's terminal. The order of transmission becomes important when several student terminals are active simultaneously. Finally, the proportion of the response that must be received at the concentrator before output to the student commences may also impose a delay on the data communications. The latter might be necessary if error correcting procedures were to be used at the line concentrator.

Following the fourth step (the transmission of information to the student), the cycle commences again at step one with the student thinking and replying.

Output Data from the Model

The model produces statistics concerning each of the four phases in the cycle. For student behavior it produces summaries of the amount of time spent in each of four postulated student activities--thinking, responding, waiting for output and reading the output. These statistics are primarily confirmatory at present, but could be used to study the system effects on student behavior.

Statistics about the second phase, queuing of transmission to the computer, concerns the length of queues associated with input to the computer and the buffer requirements within the line concentrator.

Data produced regarding the computer processing phase indicates the general load on the time-sharing system, the storage requirement and the length of input and/or output queues developed during a run. This information can then be used to estimate the load upon the central processor.

The final portion of data provided by the model concerns the transmission of information to the student. This is the portion of the data communication system that is most heavily loaded. It also represents a decrease in data transmission speed as the message approaches the student and, hence, a requirement for buffering at the line concentrator. This data is the most relevant indicator of the performance of the system as a whole. The data derived at this point include 1) the maximum and average delay from the end of an input message to the first character of output on the user terminal; 2) the number of instances of interruption of output due to non-availability of the remainder of the output; 3) the maximum buffer size required by each terminal; 4) the maximum buffer size required for all of the terminals over the time of the run and 5) the maximum length of queue on output.

The model was simulated in FORTRAN, in addition to more general queuing analyses. The remainder of this paper will be devoted to the presentation of representative results. The basic question of interest at present concerns the number of students that can be serviced by various configurations of terminal speed and concentrator data line speed. For instance, how many student terminals can reasonably be connected to the system as it exists now. That is, given terminals of the KSR 33 teletype class and a data channel of 1200 baud rate, how many students can use the system?

Assumptions

In order to explore the behavior of this CAI system it is necessary to have data regarding the behavior of several components of the system, including student behavior at the terminal, amount of processor time required per interaction, characteristics of the curriculum, etc. If data are not available, reasonable assumptions must be made until more adequate observations can be obtained. The latter is the case to a great extent in the present situation. We are beginning to get real data about system performance, but they are not yet entirely adequate. The present discussion will therefore require a mixture of assumptions tempered, where possible, with data derived from actual use of the system.

Student response times are assumed to be distributed exponentially following the conclusion of output from the computer, with a mean response time of 30 seconds. The selection of this time represents a compromise between observation and guess. Observations of students now using the remedial mathematics curriculum indicate that the mean is more nearly 60 seconds, which seems inordinately long for interactive CAI. The difference is due to the particular curriculum-student combination; the mathematics lends itself to long periods of calculation between question and answer. Calculations based upon the mean of 30 will, presumably, err on the side of safety and give ample room for future alterations of the curriculum that yield shorter interaction intervals.

Some support comes from the extensive Plato data (Bitzer and Skaperdas, 1970) which show an average of one student request every 2 to 4 seconds. A "request" is a character; when aggregating characters into 5 to 15 character responses typical of this curriculum the 30 second average is in the same range.

Computing time is assumed to be exponentially distributed with a mean of 250 msec. To this is added a constant of 200 msec. Timing experiments with portions of the mathematics curriculum have demonstrated that this is approximately correct for the current CAN interpreter. This processor overhead is exorbitant due to its being written in FORTRAN; it is estimated that the CPU cost of interpretation of CAN can be decreased by a factor of 5 to 10 by a more efficiently coded interpreter. The present report must be regarded as applying to the research and development tool currently in use; the processor utilization figures could be divided by at least 5 for any production implementation. This improvement would shorten the response time predicted for the present system.

For purposes of the model, the output messages to the student are assumed to be a mixture of instruction and problems. The former are assumed to be about 5 lines long and the problems, 1 line long. A line averages 50 characters in length. The mixture is 80% problems, 20% instruction.

The data communications assumptions of the model include the use of standard teletype terminals and an inter-computer data link operating at 1200 baud. Other speeds of terminals and link are also explored.

Results

Table 1 gives the delay in seconds between the student's carriage return and the first character of the computer's reply, in terms of average delay and the longest delay encompassing 90% and 95% of the cases. The figures given are for the teletype terminals with up to 32 students on a 1200, 1800, and 2400 baud inter-computer line. It should be noted that these figures do not include a delay due to queuing for execution at the CPU in competition with other time-sharing jobs. This is a real concern, of course, but this delay is not presently included. Any queuing resulting from this job is included, however.

Simulation results for up to 32 terminals are given. Assuming that a 2-second response time is acceptable, the experimental system is capable of providing for the 32 terminals, although better service would be obtained at higher speeds.

TABLE 1: Simulated delays in seconds between
student response and computer output

110 Baud	1200 Baud			1800 Baud			2400 Baud		
Terminals	Avg.	90%	95%	Avg.	90%	95%	Avg.	90%	95%
2	1.02	1.31	1.46	.83	1.08	1.26	.74	1.04	1.22
8	1.04	1.48	1.65	.85	1.21	1.35	.75	1.12	1.29
14	1.08	1.59	1.82	.84	1.25	1.44	.72	1.06	1.27
20	1.08	1.60	1.81	.83	1.26	1.47	.76	1.18	1.44
26	1.09	1.60	1.82	.84	1.30	1.52	.75	1.18	1.39
32	1.15	1.75	2.02	.86	1.33	1.54	.76	1.22	1.46
300 Baud Terminals									
2	1.01	1.34	1.51	.81	1.16	1.30	.73	1.02	1.18
8	1.02	1.42	1.55	.83	1.18	1.34	.74	1.06	1.20
14	1.04	1.47	1.72	.84	1.25	1.46	.74	1.10	1.32
20	1.11	1.63	1.92	.87	1.35	1.56	.75	1.13	1.36
26	1.11	1.68	1.94	.86	1.31	1.55	.77	1.22	1.45
32	1.19	1.89	2.15	.88	1.39	1.61	.78	1.28	1.52

Given the average message length and number of messages per hour, the result is about 100 characters per minute per terminal. A 1200 baud line is capable of 7200 characters per minute. Thus for 32 terminals the output portion of the line has an average line utilization of 44% From queuing theory, we can expect transmission times which include queuing to average between 1.4 and 1.8 times the actual transmission time. The 44% utilization is considerably below the critical 70 to 80% point at which queues grow very rapidly; thus it offers a considerable margin of safety. (Martin, 1972)

Alterations of the terminal configuration have the predictable consequences: using 30 character per second terminals results in increasing the rate of interaction by about 6 to 10%. This is a result of the assumption that students do not process the question or instruction until it has entirely arrived at the terminal; to some extent this is probably true and the higher speed terminals may speed the rate of instruction. Beyond that, at 30 cps we find instances of interruption of output to a terminal because the next line of information has not arrived at the concentrator when the terminal's buffer is empty. For the 1200 baud system, the interruptions occur in 3% of the messages for 14 terminals. They increase to 14% of the messages for 32 terminals. The average duration of interruption ranges from 0.1 second for 14 terminals to 0.5 second for 32. These interruptions disappear with a 2400 baud line.

Conclusion

Preliminary analyses of the data communications behavior of a medium scale CAI system based on a remote concentrator and use of a time-shared computer facility shows that the concept is feasible for up to 32 low speed terminals operating on a voice-grade data connection. The simulation and other analyses omit the factor of general loading of the time-sharing system, which could become critical to the maintenance of response time. The present interpreter places severe computing demands upon the timesharing system--up to 30% utilization for 32 terminals; this is a result of inefficiencies tolerated in a processor for research and development purposes. This should be reducible to a load of 3 to 6% in a production system.

References

Bitzer, D., and Skaperdas, D. The economics of a large-scale computer based education system: Plato IV. In Holtzman, W.H. (Ed.) *Computer assisted instruction, testing, and guidance*. New York: Harper and Row, 1970.

Churchill, S., Naess, L., and Olivier, W.P. CAN-4, an advanced author language for CAI, computer-based testing and psychological experimentation. *DECUS Proceedings*, Spring 1970, 211-217.

Martin, J. *Systems analysis for data transmission*. Englewood Cliffs: Prentice-Hall, 1972.

LES APPLICATIONS DE L'ORDINATEUR À L'ENSEIGNEMENT À L'UNIVERSITÉ DE LOUVAIN

A. Jones

INSTRUCTIONAL APPLICATIONS OF THE COMPUTER AT THE UNIVERSITY OF LOUVAIN

SESSION 3
ROOM/SALLE
148

By/Par: A. Jones,
Université Catholique de Louvain,
Belgium.

The utilization of the computer in education sets up an important problem:
- All media used in education are, in principle, good. But then, what is the right place of the computer in this panel?

There are many ways to use the computer:
- to teach
- to evaluate
- to aid.

To teach:
The computer may be used in the following modes:
- giving basic informations: we have some critics about this way of use.
- Simulation: to do this, the computer is a very useful mean. But we need another mean to give the basic informations. Good simulation asks for imaginative work.
- Heuristic research: but this supposes basic informations given by other means.
- Questions and answers: Information being given, the computer is used to correct wrong assimilation of concepts.

To evaluate:

Evaluation belongs to student and to content. This supposes that we have defined the goals of our teaching on different aspects. What actually exists is of no great utility for evaluation by computer. Evaluation of students calls for adaptative content and "Human Operator" model.

To aid: as a calculation device.

LES APPLICATIONS DE L'ORDINATEUR À L'ENSEIGNEMENT À L'UNIVERSITÉ DE LOUVAIN

L'utilisation de l'ordinateur dans l'enseignement pose un problème important:
- Tous les moyens d'expression utilisés dans l'enseignement sont, en principe, bons. Mais alors, où doit se situer l'ordinateur dans ce cadre?

Il y a bien des manières d'utiliser l'ordinateur:
- pour enseigner
- pour évaluer
- pour aider

Pour enseigner:
On peut se servir de l'ordinateur pour:
- donner des renseignements fondamentaux; cette façon d'utiliser l'ordinateur peut se prêter à quelques critiques.
- simuler. L'ordinateur est un moyen très utile mais il faut un autre moyen pour donner les renseignements fondamentaux. Une bonne simulation exige un travail d'imagination.
- faire des recherches heuristiques mais ceci suppose des renseignements fondamentaux fournis par d'autres moyens.
- poser des questions et avoir les réponses. Les renseignements étant donnés, on utilise l'ordinateur pour corriger une mauvaise compréhension.

Pour évaluer:

L'évaluation appartient à l'étudiant et au contenu. Ceci suppose que nous avons défini les objectifs de notre enseignement sous ses différents aspects. Ce qui existe actuellement n'est pas d'une grande utilité pour l'évaluation par l'ordinateur. L'évaluation des étudiants implique une souplesse d'adaptation de la machine qui doit obéir à un modèle d'"opérateur humain".

Pour aider: comme calculateur.

COMPUTER ASSISTED INSTRUCTION ON A SMALL COMPUTER

BY

ANN BREBNER AND H. J. HALLWORTH

COMPUTER APPLICATIONS UNIT

THE UNIVERSITY OF CALGARY

CALGARY, ALBERTA

COMPUTER ASSISTED INSTRUCTION ON A SMALL COMPUTER

SESSION 5A
ROOM/SALLE 143

By/Par: Ann Brebner, H.J. Hallworth,
University of Calgary.

The CAI project in the Faculty of Education at the University of Calgary has been primarily concerned with the use of a small time shared computer.

Seven terminals are in use in the University, and nine are currently located in schools and a research institute for the developmentally handicapped.

The paper describes the development of simulations, computer assisted instruction and computer managed instruction programs. A CAI language and an information system have been written for the time shared machine, and are discussed.

Most programs have been developed for use at school level. They relate, for example, to elementary arithmetic, English, social studies, mathematics, physics and counselling. Particular attention is now being given to programs for the handicapped.

Experience in the use of this machine indicates certain advantages and limitations of a small computer. It is comparatively inexpensive to bring such a system into operation and develop a variety of programs. However, the difficulty of maintaining a number of programs in constant use indicates the need for access to a larger machine and a more versatile CAI language.

Particular attention is drawn to the need for terminals specifically designed for CAI, and the need to obtain some compatibility of both hardware and software within Canada in order to facilitate co-operation in the development and use of instructional programs.

L'ENSEIGNEMENT À L'AIDE D'ORDINATEURS SUR UN PETIT ORDINATEUR

Dans l'étude EAO à la faculté d'éducation de l'Université de Calgary, on s'est intéressé surtout à l'utilisation d'un petit ordinateur à temps partagé.

A l'Université, on se sert de sept terminaux et, actuellement, il en existe neuf dans les écoles et dans un institut pour les handicapés.

Dans cette communication, nous décrivons le développement de simulations, l'enseignement à l'aide d'ordinateurs et les programmes d'enseignement dirigés par ordinateurs. Un langage EAO et un système d'informations ont été écrits pour la machine à temps partagé et nous en discutons.

La plupart des programmes sont conçus pour être utilisés dans les écoles. Ils se rapportent, par exemple, à l'arithmétique élémentaire, à l'anglais, aux études sociales, aux mathématiques, à la physique et à l'orientation. À présent, on étudie particulièrement des programmes pour les handicapés.

L'expérience de l'utilisation de cette machine démontre des avantages et des limitations d'un petit ordinateur. Il est relativement peu coûteux de mettre ce système en opération et de développer des programmes différents. Cependant, il est difficile de faire marcher plusieurs programmes qui sont constamment en usage et nous constatons qu'il est nécessaire de se servir d'une plus grande machine et d'un langage EAO mieux adapté.

Nous attirons l'attention sur le besoin de terminaux conçus spécialement pour EAO et sur la nécessité d'une compatibilité entre l'ordinateur et la programmation dans notre pays afin de faciliter la coopération en ce qui concerne le développement et l'utilisation de programmes pédagogiques.

The computer assisted instruction project in the Faculty of Education at the University of Calgary has been centred round a small time-shared computer, namely a Digital Equipment Corporation TSS-8 system.

More specifically the present hardware configuration consists of a PDP-8/I computer with a 12 bit word length, a time-sharing option (KT8/I), and an extended arithmetic option (KE8/I). There are 16K words of core store; a 256K word disk (RS08) and disk control unit (RF08); six DECtape units (TU55) and a DECtape control unit (TC01); and sixteen PT08 terminal interfaces, ten with F options to allow attachment to a phone line.

The system supports seventeen terminals. Seven of these are in the University and are hard-wired to the computer. Four are in a trailer and are attached to the computer by means of a multiplexer and two phone lines. The remaining terminals are located in schools and in a research institute for the developmentally handicapped. These communicate with the computer through acoustic couplers and the ordinary city telephone network, and share the remaining available ports. This latter method is the most flexible means of communication, as by simply dialling another appropriate phone number the terminals can be attached to a different computer such as the CDC 6400 in the University Data Centre or the PDP 10 at the National Research Council in Ottawa.

Four of the terminals in the University and four terminals in the trailer are mounted on computer assisted instruction stations. Each of these is equipped with a teletype, a random access slide projector and a rear view screen. Output from the computer to a CAI station passes through a special interface. Normally information passes straight through to the teletype, but when a specific non-printing character is received by the interface, the following character is routed to the slide projector. Therefore, at these stations, material can be presented both on slides and on the terminal printer.

The software for this interface has been written in such a way that it is extremely easy for the programmer to use. He merely has to call a function, the argument of which is the number of the slide to be displayed. This slide remains on the screen until the function is called again with a different argument.

The high level languages supplied with the TSS-8 system are FOCAL, BASIC, and a version of FORTRAN; there is also an assembler language, PALD, and a set of utility programs to manipulate and transfer files between the disk and other peripheral devices. There is no CAI author language implemented on the system and consequently most CAI programs have been written in either FOCAL or BASIC. A special purpose CAI language, SCROLL, has been developed and is used for some programs.

Many students in the Faculty of Education presently learn to use the system. Courses on the use of computers in education are run both at the undergraduate and graduate level. In these, students are first taught a computer language and then how to write computer assisted instruction programs. Those who are teachers, and those who are engaged in practice teaching, are encouraged to take terminals into their schools and make use of the programs they have written.

Students in an intermediate statistics course use the terminals in two ways. Firstly they have available a set of statistical programs into which they can put their own data to obtain histograms and descriptive statistics, means, standard deviations, correlation coefficients, t-tests, chi-square tests, and one and two way analysis of variance. Secondly, they learn the elementary concepts of the FOCAL language and write their own programs for statistical techniques which are not available in the above package.

The terminals have been used during the current session to assist with the teaching of the elementary statistics section of one of the large introductory classes. This experiment took the form of a set of eight CAI programs and a self-evaluation test written in the BASIC language for use at the CAI stations. The programs were essentially user controlled in that the student could work through them in any order (although there was a suggested path). He could also work through any program as many times as he wished, and could take the self-evaluation test at any time. His answer was not always checked by the computer. Frequently he was asked to write down his answer, and check it against the correct answer printed on request by the terminal. This amount of user control appeared to work well at undergraduate level with a student group having widely diverse statistical backgrounds.

Terminals linked into the TSS-8 system have been placed in schools for the last three years and the majority of CAI programs written have been for use in schools. These range from drill and practice programs for elementary grades to calculus and physics programs for grades 11 and 12.

By far the largest group of programs are those for drill and practice in elementary arithmetic, which require only a teletype terminal. There are presently some sixty programs in this area. They give practice, at various levels of difficulty, in counting, basic number facts, place value and addition. Recently, several programs in fractions and problem solving have been developed. All programs are constantly being revised and rewritten in consultation with the staff of schools where they are in regular use.

A set of programs on addition and place value was designed specifically for use with the IPI (Individually Prescribed Instruction) project which is currently in operation in a Calgary elementary school. A management system was also developed for use in conjunction with this project. This is currently being re-written in a more general form in order that it may be used by a greater number of schools.

A set of drill and practice programs in English is under development, concerned with noun and verb agreement.

For two years the trailer has been located at a combined high and junior high school. The terminals are used regularly by students of math option classes who learn to program in the FOCAL language. CAI programs written for schools which require the slide projectors are also run in the trailer. These include a calculus program in which it is necessary to display graphs of functions; programs on problem solving in physics where diagrams and equations are shown on slides; a CAI program on map reading; and a program on vocational counselling. In this latter program the slides are used to display to the student first a list of interests, then a list of aspired educational levels. The student chooses one from each list and is presented with a matching list of occupations, again on a slide. He may then choose one or more of these and details of the job are printed out on the teletype. No attempt is made to match his choices with his present achievements except to list for him the further courses and grades he needs in order to prepare himself for the chosen job.

Other programs which are occasionally run in schools are simulations and games. Among this group the more educational programs are on elementary economics, fur trading in Quebec in the 1760's, farming in Kansas in 1880, and ranching in Alberta. The objective in all these is essentially the same, to keep the enterprise running from year to year. Other programs simulate biological systems, for example an aquarium, or experiments in physics. Most of the latter have been developed in co-operation with Dr. C. Anger of the Physics Department, and are now being implemented on a CRT terminal with simple graphics.

More recently, CAI programs have been written for use in an institute for the developmentally handicapped. Much of the instruction carried out in such an institute is necessarily done on an individualized basis due to the widely varying abilities of the trainees. Some of the drill and practice programs written for use in the elementary schools can equally well be used here; other programs are those concerned with everyday living. The latter include coin recognition, coin equivalences, making change and budgeting. There is also a set of programs designed to help trainees recognise the tools used in job training courses.

When the slide projectors were successfully interfaced to the computer, a whole new range of possibilities for CAI programs opened up. For example, until this time it had been difficult to produce many programs for mathematics because it is not possible to present algebraic equations in the correct form on a teletype. Programs also became feasible on such topics as the recognition of alpine flowers or constellations of stars.

Further technical developments in terminals for CAI work again greatly widen the scope of effective programs which can be written. These include the ability to play computer controlled audio messages and to accept input to the program through the NRC touch sensitive display. Such terminals may be used for spelling programs, for example, and by that group of students who cannot read a written message or type a reply on the keyboard.

It therefore appears that there is a need for a complete range of terminals, from the simplest type with printer and keyboard for the teaching of programming, to the more complex type for the running of CAI programs.

More important, however, is the need for compatibility of codes, particularly the non-printing control codes, in order that programs can be readily transferred between terminals. For CRT type terminals, and for terminals attached to a small computer, there is also a need for some sort of local storage facility.

The greatest disadvantage of running a CAI project on a small stand-alone time-shared computer is the lack of sufficient storage space for programs and data. The point is rapidly reached where there is need for tight scheduling of programs stored on the disk and much time is spent transferring programs between the disk and other peripheral storage devices. Another disadvantage is that the number of terminals which can be attached simultaneously to the system is relatively small: the seventeen terminals attached to the system described above is the maximum number for this configuration. The fact that there is no CAI language as such has not proved a serious disadvantage. With a little ingenuity on the part of the programmer, effective CAI programs can be written in BASIC and to a limited extent in FOCAL. This does, however, mean that the development of CAI requires a relatively skilled programmer.

An important advantage of the small computer is that it can be dedicated entirely to CAI work. Also, a small machine is relatively inexpensive and can be built up gradually, starting if necessary with the basic computer, 4K of core and one teletype. It is therefore a viable system for a school board or group of schools.

Of the programs which have so far been developed, simulations make good use of the computer, are popular with the students and could be used to teach a substantial part of the curriculum. For example, the simulation of the various aspects of the operation of a company has a useful place in a business education course, and could possibly be made the basis for such a course. Simulations can also play a major part in a physics or chemistry curriculum. Drill and practice programs are easily written, but there is need for careful sequencing of the programs and for different levels of difficulty for each topic. They should be organized so that any one in the set is easily called into use by the teacher and, if possible, they should include some automatic logging of the students' performance. Similarly, any management system must be very easy for the teacher to use: it must aim to reduce the teacher's routine work and allow him time to help students with problems which cannot be resolved by a CAI program.

Large branching types of tutorial programs are not practicable on a small computer. However, limited programs of this type can be written if there are terminals on the system which can present material on slides or audio tapes.

The areas to which CAI can be best applied at the present time, and in the immediate future, are those in which the ratio of students to teacher is small. These are the areas in which CAI is already becoming an economic proposition. They include remedial work, counselling, and instruction in institutes for the developmentally handicapped.

It is considered that the programs described in this paper have demonstrated the practicability of computer assisted instruction on a small machine. As the point is reached where larger sets of programs are being produced, sufficient to cover a greater part of the curriculum, there is need for a careful and critical evaluation of the educational value of different types of programs. Also, as we approach the stage at which programs are written which will be of considerable relevance in the school classroom and at university level, there is a large task ahead in producing the appropriate educational materials.

It is apparent that other groups in Canada will be engaged in a similar task. It is equally apparent that increasing co-operation would be to the advantage of all.

The first steps in this direction require the development of compatible hardware and software and, insofar as possible, access to a large central computer on which programs can be tried and exchanged.

These are the objectives of the N.R.C. computer-assisted instruction project which is described in the following paper.

ACKNOWLEDGMENTS

Work reported in this paper has been supported in part by The National Research Council of Canada, Canada Council and the Alberta Human Resources Research Council.

The N.R.C. Computer Aided Learning Cooperative Research Project

J. W. Brahan & W. C. Brown
National Research Council of Canada

THE NRC COMPUTER AIDED LEARNING COOPERATIVE RESEARCH PROJECT

SESSION 5B
ROOM/SALLE 143

By/Par: J. W. Brahan, W.C. Brown
National Research Council of Canada

In 1967, the National Research Council began a preliminary study of the application of computers as aids to learning. This initial work led to the establishment of a central research facility which is used by NRC and a number of educational research organizations in a cooperative programme of research into Computer Aided Learning. This central facility includes a medium scale time-sharing computer which is accessible to the participating organizations by means of remote terminals.

A major objective of the project is to provide a facility which will allow the active cooperation of research workers throughout Canada in the development and evaluation of Computer Aided Learning systems to meet Canadian requirements. The NRC efforts are concentrated in the areas of development of terminal equipment, specialized computer facilities, and system programmes. Examples of such development include audio tape and disc storage units, a transparent touch sensitive tablet for computer input, alphanumeric and graphic display devices, line concentrator systems and supporting system programmes. The cooperating educational organizations provide the facilities for the development and testing of course material and evaluation of terminal equipment and in some cases, assistance in the development of system programmes.

The first participating educational organization to go "on line" was the Ontario Institute for Studies in Education in Toronto which was linked to the computer early in 1970. Since that time, the number of on line participants has grown to five.

The project provides a means for close communication between the educational researcher and the system designer. In this way, effective use can be made of available resources to arrive at a system specification which more closely meets the requirement of the user than might otherwise be possible.

LES RECHERCHES FAITES EN COLLABORATION AVEC LE CNRC

En 1967, le Conseil national de recherches a entrepris une étude préliminaire sur l'application des ordinateurs pour faciliter l'enseignement. Ces premiers travaux ont conduit à l'établissement d'une installation centrale de recherche qui est utilisée par le CNRC et d'autres organismes de recherches pédagogiques dans le cadre d'un programme coopératif de recherches sur l'enseignement à l'aide d'ordinateurs. Cette installation centrale comprend un ordinateur à temps partagé de taille moyenne qui est accessible aux organismes participants au moyen de terminaux éloignés.

Un des principaux objectifs de l'étude est de fournir une installation qui permettra la coopération active des chercheurs dans tout le Canada pour développer et évaluer des systèmes d'enseignement à l'aide d'ordinateurs satisfaisant les conditions canadiennes. Les efforts de CNRC portent surtout sur les domaines du développement d'équipements terminaux, d'équipements spécialisés et de programmes de systèmes. Voici quelques exemples de ce développement: des mémoires audiophoniques, sous forme de bandes magnétiques et de disques, un écran d'entrée sensible et transparent, des dispositifs d'affichage graphiques et alphanumériques, des systèmes de concentration des lignes de communication et des programmes appropriés. Les organismes pédagogiques coopérants fournissent les installations pour développer et essayer le contenu des cours et pour évaluer l'équipement des terminaux et, dans certains cas, ils aident à développer les programmes adaptés à des systèmes particuliers.

Le premier organisme pédagogique participant, à être en liaison directe, est l'Institut ontarien de recherche en éducation à Toronto qui a été relié à l'ordinateur dès 1970. Depuis cette époque, le nombre de participants en liaison directe est monté à cinq.

L'étude donne un moyen de communication rapprochée entre le chercheur pédagogique et l'ingénieur ayant conçu le système. De cette façon, on peut utiliser les ressources disponibles d'une manière efficace pour arriver à des normes qui satisfont mieux les besoins de l'utilisateur.

In 1967, the National Research Council began a preliminary study of the application of computers as aids to learning. It was immediately apparent that effective work in this field could not be carried out without the active cooperation of the educator. Thus very early in the project, contact was established with the University of Calgary and the Ontario Institute for Studies in Education in Toronto. Both of these organizations have been actively participating in the project since its early stages and have significantly affected the direction which it has taken.

The N.R.C. program is aimed at the development of facilities necessary to permit the computer to be used effectively by student, teacher, and researcher for conversational tutorial interchange, drill and practice, enquiry operations for information retrieval from libraries and data banks as well as the entensive computational operations normally associated with the computer. Additionally, and perhaps most important is the use of the computer to simulate experiments which might be too lengthy, costly or dangerous to be carried out in the laboratory, or to simulate situations which result in the student experiencing responsibility through realistic decision making.

Acceptance of computer aided learning by students, teachers and others will be dependent on the availability of a wide choice of fully evaluated courses, the adequacy of the terminal and system facilities for the presentation of subject matter and acceptance of user input, and the degree of equipment and communication standardization which can be achieved in the interest of cost reduction. The terminal and system facilities must provide the user, be he student, teacher or researcher with full access to the power available to him in the computer. Well defined measures of the benefits to be gained from computer aided learning must be available to the educational administrator to permit him to weigh the advantages and costs involved in order to decide when, where, and how to implement CAL systems.

To provide a means whereby effective use could be made of available resources in the research and development effort required to arrive at an acceptable computer aided learning system, the National Research Council established a central research facility aimed at catalysing a national effort in this field. To encourage the free exchange of techniques, course materials, and evaluated results necessary to achieve the required degree of standardization, N.R.C. services in systems programming, special equipment design, and computer time are offered without charge to collaborating groups undertaking research projects to further the overall aims of the program. N.R.C. undertakes no curriculum development; this is contributed by the educational research groups engaged with us in this program.

Some of the advantages offered by the centralized research facility include:

- Orderly evolvement of necessary standards
- Possibility for immediate exchange of programmes and course materials at the research stage through the "electronic publication" facilities of the computer
- Availability of a system for support of specialized author languages and testing of experimental features by the participating organizations in a wide range of applications.

The provision of adequate facilities to permit interested groups to undertake research in common with the goal of defining system requirements is one of the main objectives of the N.R.C. program. Thus the design and technical evaluation of specialized computer terminal equipment and system programmes to permit its use in a flexible manner are features of the development work underway in the laboratories of the Information Science Section.

Work in the area of student terminal development is concentrated on input, storage and display aspects and on the development of the software required to make effective use of the features incorporated in these specialized terminals through the central computer facility. These efforts are aimed at the development of terminal facilities which are not currently available and at their integration with equipment which is available. A major emphasis is placed on conformity to applicable standards. Some examples of the work on specialized facilities for CAL terminals include the following:

- A transparent touch sensitive input tablet (1) has been developed which will transmit data to the computer when the surface of the tablet is touched. This device uses ultrasonic surface waves on glass, transmitted sequentially from two adjacent edges, to determine the position of the finger or other passive stylus such as a pencil. The tablet is being produced commercially under license from N.R.C.

- Methods of character generation are being investigated giving due regard to the requirements of displaying text, in both English and French, of sufficiently high quality that there is no eyestrain. Methods are also being investigated for the storage and display of graphical information, both line drawings and T.V. still pictures with grey scale in conjunction with alphanumeric displays.

- To permit the presentation of audio information, experimental equipment has been developed which permits the computer to select pre-recorded messages from a conventional audio tape recorder. An "instantaneous response" audio unit is currently under development which will eliminate many of the problems in tape storage of information, notably the long delays which are sometimes encountered in retrieving a desired message.

Special purpose components such as the above have been assembled with a conventional keyboard-printer computer terminal and the necessary communications interface designed to permit operation of this multi-media student terminal from the central computer through a conventional telephone connection.

With the aim of achieving more economical access to the computer, programmes have been developed for small multiplexing computers, which when used at local centres, permit a number of simple and/or complex student terminals to share a common communication channel to the central computer. Methods of sorting and servicing messages arriving on such shared channels by the central computer are being investigated. System programmes involving two quite different approaches have been developed for this purpose.

Software techniques are also being investigated to improve the effectiveness of student operation of input devices. One example of such a technique is a programme which can correctly identify words containing spelling errors.(2) This programme takes into account the possibility these may have been introduced as a result of a "keyboard error" caused by striking the wrong key, or possibly a transient equipment malfunction, or as a result of human error - the user not knowing how to spell the word correctly.

Other research which is expected to lead to additional computer facilities in the future is directed towards achieving voice input to computers and economically feasible mass storage methods for use with library and curriculum materials. Work on storage devices is currently focussed on magnetic and semiconductor thin films.

The N.R.C. program has, as one of its objectives, the consistent classification, indexing, storage and retrieval of CAL curriculum and conventional library materials by means of the computer. Thus the student, teacher or author will be able to manipulate and retrieve material, using natural language commands, as an extension of the learning process. Consistent retrieval by computer, once classification and indexing are adequately achieved, is not too difficult; the greater problem lies in assigning classification and index terms automatically without human intervention.

The techniques used to accomplish automatic classification and indexing are statistical in nature but word counting is not used. For a given category, established either by human or machine means, a word list is produced automatically with appended probability values that the word belongs in that category. A document is indexed by testing its words against the words of each such category list, the word probability values being summed within each category for each match. The document is directed to the classification wherein the maximum sum is found. This approach permits the value attached to each word to vary with time as its usefulness as an index term changes and also allows the continuous updating of the word list as new terminology enters any established category. The uniqueness of the terms to a given category determines the probability value automatically assigned. The techniques used are independent of meaning and language; retrieval requests may be formulated naturally in the language of the document to initiate its retrieval, for the request is similarly analysed to allow matching of word lists. The output of the search is ordered as to anticipated usefulness and the decision threshold is adjustable. Production of abstracts by computer is possible with the technique employed and the automatic correction or assignment of classification is now under investigation.

A strong emphasis is placed on standards throughout the program. Existing standards are adhered to wherever possible, and where existing standards do not adequately meet the requirements, or no standards exist, representation is made to appropriate standards groups. It is felt that only by the definition of and adherence to acceptable standards can results of this work be made available on a widespread basis. Two areas in which it is particularly important that standards be achieved soon are those of programming languages for CAL applications and communication codes for use in Canada to permit efficient exchange of information between computer systems in both English and French.

The medium scale time sharing computer, a Digital Equipment Corporation PDP-10, which forms part of the N.R.C. central research facility, provides a means whereby remotely located research groups can actively participate in the CAL project. Educational organizations throughout Canada can develop and evaluate course materials, evaluate system facilities, both hardware and software, and contribute to the definition of system parameters. The sharing of a central computer during the research phase permits course materials to be readily transferred from one participating organization to another. The storage of course materials and programmes in on-line, machine readable files which are accessible to all participants means that there is little or no delay in the transfer of information from one group to another. Thus provided the originator of the material has supplied sufficient documentation in the file, the recipient can make use of the material with very little delay. In this way, through the cooperation of several participating organizations, it is possible to test a programme more extensively and under more varied conditions than if the testing was done by a single organization working in isolation.

Currently five organizations have projects underway which make use of the N.R.C. computer facility and others are expected to join the system later in the summer.

The projects currently on the system are illustrated in Fig. 1.

The first participating educational organization to go on line was the Ontario Institute for Studies in Education in Toronto which was linked to the Ottawa computer early in 1970. Initially this link terminated in Toronto at a single teletype which was used primarily for programme development. Now at the Toronto end of the line is a small time shared computer which serves as a line concentrator. By means of this facility, a number of individual student terminals connected to the line concentrator in Toronto can share a common communications link to the Ottawa computer. Currently twelve terminals are linked through the O.I.S.E. computer in this way and consideration is being given to increasing this number.

One phase of the current O.I.S.E. activity(3,4) represents a major contribution to the system facility. This consists of the development and testing of a course-authoring language interpreter, CAN6, along with the development of analysis routines for processing data gathered during experimental applications of the language. Entensive evaluation of the CAN6 language and system facilities is being carried out as a result of the development and testing of a computer-based remedial mathematics curriculum which the O.I.S.E. is undertaking in cooperation with a number of Ontario Community Colleges. As part of this project, there is a direct link from the N.R.C. computer to Algonquin College in Ottawa as well as the indirect links through the O.I.S.E. to several colleges and high schools in the Toronto area.

At Collège Edouard-Montpetit, in Longueuil, a project is underway which is aimed at the evaluation of the characteristics of the computer programming language LOGO,(5,6) which was developed by Bolt Beranek and Newman in cooperation with MIT and several other educational organizations. As part of this project, a French language version of LOGO is under development, as well as a line concentrator programme for a PDP-11 computer which will provide a number of student terminals access to the LOGO facility through a common telephone line to the N.R.C. computer.

As noted earlier, the University of Calgary has been cooperating with N.R.C. in the CAL work since the early stages of the project. The initial contacts between the groups were established in late 1967. However, connection to the Ottawa computer facility was not made until late in 1971 and is on a quite restricted basis because of the high costs of communication. Extensive work has been underway at Calgary for some time however using a small time shared computer.(7) As part of the Calgary project, an experimental student terminal developed at N.R.C. has been supplied for evaluation. This terminal consists of a slide projector with touch-sensitive screen, an audio tape recorder and a conventional keyboard-printer unit, all capable of being operated under computer control through a single telephone connection to the computer. One of the objectives of the cooperative project with the University of Calgary is an investigation of the communication and other problems associated with the use of multimedia student terminals. Of particular importance is the investigation of the problems associated with communicating over long distances to such devices using common carrier facilities.

At McMaster University in Hamilton, an initial investigation of CAL is currently underway as part of a study of science concept learning at university level. One of the facilities installed in a resource centre used in the study is a terminal linked to the N.R.C. computer. One of the objectives of the project is to investigate methods of integrating a variety of learning resource materials with more conventional methods. Thus the resource centre contains, in addition to the computer terminal, a variety of audio-visual facilities aimed at providing an individualized approach to science teaching.

At N.R.C. last year, an in house project was started, aimed at identifying and developing the necessary facilities in a CAL system for satisfying the special requirements of students with learning disabilities. The initial work undertaken on this project has been the computer implementation of the Peabody Picture Vocabulary Test which yields an IQ score. The implementation of this test, which was carried out in cooperation with the Psychology Department of Carleton University, makes extensive use of the special facilities provided in the experimental student terminal developed at N.R.C. It is expected that this initial work will be the first phase of an automated perceptual screening project to be undertaken by Carleton University's Department of Psychology.

Project proposals from the Université de Montréal and the University of Western Ontario have been received and it is expected that research groups at these centres will be linked to the N.R.C. computer when details have been worked out.

Projects which make use of the central research facility must contribute to the achievement of the overall goals of the cooperative program. This contribution is in the form of helping to define and evaluate system characteristics, either hardware, such as special purpose terminal equipment, or software, such as special purpose languages and other software available for system support. It is not the purpose of the program to provide a tool for purely educational research. The research must be related to the development of CAL facilities and techniques. Thus the focus of contact between the N.R.C. group and the cooperating educational organizations is the technology and the individual projects must reflect this. The use of the computer is furthermore restricted to the research phases of the project. Once a project becomes operational and is to be run as a day to day service with the results of such operation not being used for research, it is required that other facilities be used.

Use of the central computer facility has been building up over the past year and the benefits of the central facility are being demonstrated. As the project continues and problem areas are identified and resolved it is to be expected that use of the central facility will become considerably more effective over the next year.

In closing, it should be noted that the interest of educational research organizations in participating in the cooperative project is much greater than the number of users of the system would indicate. However, the high cost of communications is at present a major deterent to potential participants who are located at any great distance from Ottawa. To make the program truly cooperative, it is our hope that equalized communication charges, independent of distance can be arranged. It is hoped that the use of a central computer research facility will prove its worth in CAL research and development and thus encourage similar programs of sharing the study of other multidisciplinary problems through the use of modern communications facilities.

References

1. Hlady, A. M. - "A Touch Sensitive X-Y Position Encoder for Computer Input" Proceedings of the Fall Joint Computer Conference, Vol. 35, 1969.

2. Symonds, Martha - "Computer Detection of Misspelled Words" National Research Council of Canada, Radio and Electrical Engineering Division Report ERB 843, August, 1970.

3. McLean, R. S. - "A Model of a Centralized CAI System" Proceedings of the Canadian Symposium on Instructional Technology, May, 1972.

4. Olivier, W. P. - "Computer-Assisted Mathematics for Upgrading College Students' Skills" Proceedings of the Canadian Symposium on Instructional Technology, May, 1972.

5. Feurzeig, W. et al - "Programming Languages as a Conceptual Framework for Teaching Mathematics" Bolt Beranek and Newman Inc. Report No. 2165 Submitted to National Science Foundation, 30 June, 1971. Contract NSF-C4615.

6. Papert, S. and Solomon, Cynthia - "Twenty Things to do With a Computer" Educational Technology Vol. 12, No. 4, April, 1972.

7. Brebner, Ann and Hallworth, H. J. - "Computer Assisted Instruction on a Small Computer" Proceedings of the Canadian Symposium on Instructional Technology, May, 1972.

NATIONAL RESEARCH COUNCIL

COMPUTER AIDED LEARNING PROJECT

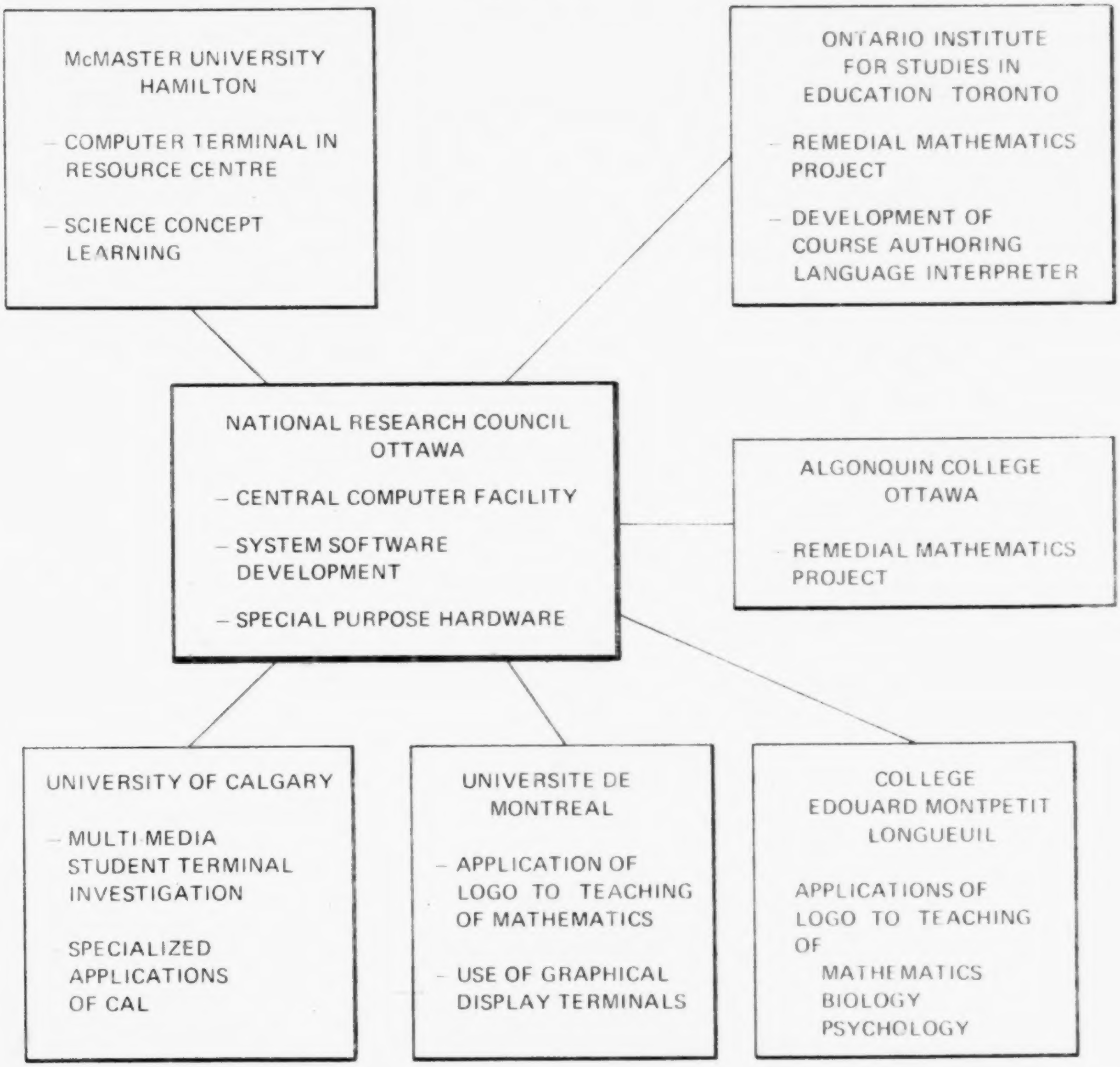

McMASTER UNIVERSITY
HAMILTON
– COMPUTER TERMINAL IN RESOURCE CENTRE
– SCIENCE CONCEPT LEARNING
ONTARIO INSTITUTE FOR STUDIES IN EDUCATION TORONTO
– REMEDIAL MATHEMATICS PROJECT
– DEVELOPMENT OF COURSE AUTHORING LANGUAGE INTERPRETER
NATIONAL RESEARCH COUNCIL
OTTAWA
– CENTRAL COMPUTER FACILITY
– SYSTEM SOFTWARE DEVELOPMENT
– SPECIAL PURPOSE HARDWARE
ALGONQUIN COLLEGE
OTTAWA
– REMEDIAL MATHEMATICS PROJECT
UNIVERSITY OF CALGARY
– MULTI-MEDIA STUDENT TERMINAL INVESTIGATION
– SPECIALIZED APPLICATIONS OF CAL
UNIVERSITE DE MONTREAL
– APPLICATION OF LOGO TO TEACHING OF MATHEMATICS
– USE OF GRAPHICAL DISPLAY TERMINALS
COLLEGE
EDOUARD MONTPETIT
LONGUEUIL
APPLICATIONS OF LOGO TO TEACHING OF
MATHEMATICS
BIOLOGY
PSYCHOLOGY

FIGURE 1

Computer-Assisted Mathematics for Upgrading Student Skills

W. P. Olivier

The Ontario Institute for Studies in Education

COMPUTER-ASSISTED MATHEMATICS FOR UPGRADING COLLEGE STUDENTS' SKILLS

SESSION 5C
ROOM/SALLE 143

By/Par: Dr. W. P. Olivier,
Institut ontarien de recherche en éducation.
The Ontario Institute for Studies in Education.

The Ontario Institute for Studies in Education (OISE) and several Colleges of Applied Arts and Technology (CAATS) jointly are developing a curriculum for upgrading mathematics skills. The project uses the computer facilities of the OISE and the National Research Council for on-line diagnosis and instruction of mathematics skills. The Computer-Assisted Learning system software was designed at the OISE and implemented with a view toward independence from any specific computer; therefore all system programs are coded in widely standarized languages. The authoring language, CAN-4, which is interpreted by the system programs, allows the curriculum designer and student a large degree of flexibility in the instructional process.

The mathematics curriculum material is designed as a set of modules keyed to behavioral objectives prerequisite for first year CAATS students. The project's goal is to provide the student with the most efficient means of instruction suited to his individual needs. Advanced assessment techniques requiring an on-line computer system provide valid decisions which can reduce, by seventy percent, the student's time required for mastery of the curriculum. Reports on the performance of each student and course are used by the project staff to improve the system and the teaching/learning process.

CONTRIBUTION DE L'ORDINATEUR POUR ÉLEVER LE NIVEAU EN MATHÉMATIQUES DES ÉTUDIANTS DES COLLÈGES LOCAUX

L'Institut ontarien de recherche en éducation, en collaboration avec plusieurs collèges de technologie et des arts appliqués (CAAT, en anglais), élabore un programme pour élever le niveau des étudiants en mathématiques. Ce programme est basé sur les installations de l'Institut ontarien de recherche en éducation et du Conseil national de recherches pour les diagnostics et l'instruction en mathématiques par liaisons immédiates. La programmation pour le système d'instruction à l'aide d'ordinateurs a été conçue à l'Institut ontarien de recherche en éducation et a été développée en vue d'être indépendante de tout autre ordinateur particulier; donc tous les programmes sont codés en langages bien connus. Le langage d'injection, le CAN-4, qui est interprété par les programmes, donne à l'enseignant et à l'étudiant des moyens de se servir de méthodes très souples d'instruction.

Le programme des mathématiques a la forme d'un ensemble de modules basé sur le comportement que l'on veut obtenir des étudiants de première année des CAAT. Notre but est de donner à l'étudiant les moyens les plus efficaces d'instruction dans son cas particulier. Des techniques avancées d'évaluation qui nécessitent une liaison immédiate avec un puissant ordinateur, permettent de prendre des décisions valables qui peuvent réduire de soixante-dix pour cent le temps dont l'étudiant a besoin pour maîtriser le programme. Des rapports sur les résultats obtenus par chaque étudiant et sur le cours lui-même servent aux expérimentateurs à améliorer le système et le processus d'enseignement et d'assimilation des connaissances.

How often have you listened to speakers and read papers predicting the wonderful benefits of computer-assisted instruction (CAI)? This paper will not be concerned with the future of CAI. The computer, with sophisticated instructional and testing techniques currently available, can minimize the amount of information passed between computer and student. This minimal information transfer thus increases the efficiency of the learning process. The project which is reported here should give indications of the benefits of computer-assistance for a specific curriculum area with techniques and facilities available today.

The Individualization Project

The Individualization Project of the Department of Computer Applications uses evolutionary, not revolutionary, methods to provide students with the most efficient instruction suited to his individual needs. Our project is a coordinated, multi-facetted approach to the problem of individualizing instruction through the use of a computer.

Why use a computer? Simply because it is the only general-purpose control mechanism which is capable of the tremendous information-processing burden demanded by the individualization of instruction. Unfortunately, we know more about the designing of computer systems than we do about the ways of maximizing learning, retention and transfer while minimizing the student's time spent learning.

The Individualization Project has designed CAI systems for small general-purpose timesharing computers and for dedicated multi-processor systems capable of processing hundreds of students concurrently. The computer systems (Churchill, Naess & Olivier, 1969) have software for interpreting a special purpose course authoring language called CAN-4, registering students to use courses, checking language syntax of CAI programs and producing reports on student and curricular performance. Another component of the project accesses the suitability of available computer-student interface devices and designs such hardware when none is available commercially. Research projects are also being conducted on individual learning differences encountered by the individualization of instruction. Last, but not least, a large scale mathematics curriculum project is under development and showing extremely good results during the initial field trials.

The benefits of working in such an environment with the cross-fertilization of ideas is enormous. In almost all cases, the system software and hardware is modified to enhance the utility of the curriculum software (courseware), rather than the usual situation of restricting the courseware development.

The major focus of this paper is to report on the development of a CAI curriculum for community college mathematics in Ontario. This sub-project is the major focus of our Individualization project; therefore, the other components will be mentioned only briefly, where they contribute to the mathematics sub-project.

The first thought that probably enters one's mind is "Oh, I've seen many CAI math projects. Isn't the OISE just reinventing the wheel?" We think not. We hope that you perceive the benefits of our work by the conclusion of this paper.

Needs and Goals

Over several years and from several sources, such as the Ontario Mathematics Commission, it has been reported that a large proportion of students entering the community colleges are deficient in mathematical skills. A mathematics deficiency means that students lack the knowledge of mathematics necessary to pursue effectively their professional courses. The mathematical deficiencies are not limited to specific topics in mathematics. As a result of the demonstrated societal and institutional needs for a mathematics curriculum catering to individual differences, the OISE and several community colleges began development work with the following goals.

1. To fill the need for high quality, original, individualized deficiency diagnosis of mathematical skills at the community college level of instruction,
2. to provide individualized remedial instruction in these skills; and,
3. to do so at a cost which is acceptable to the agencies providing the students with this instruction.

Project Structure

Although the OISE initiated the project, it is the users, the community colleges, which determine the interim goals and means for attaining these goals. Figure 1 shows the project structure. The Deans' Policy Committee is responsible for committing college resources to the project. The committee is composed of one member from the OISE and one Dean from each of the participating colleges. The Project Committee is composed of OISE staff, and teaching staff from the colleges associated with the project. This committee creates the mathematics skills specifications, designs tests to measure these skills and writes instructional sequences to teach the students. The Editorial Board is composed of one course author from each college

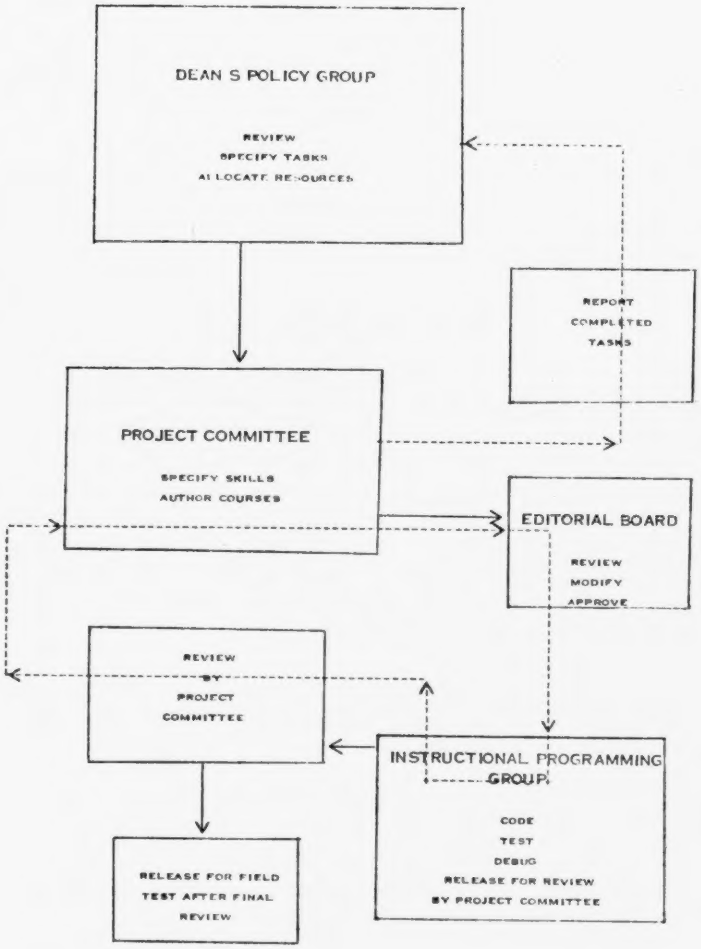

DEAN S POLICY GROUP
REVIEW
SPECIFY TASKS
ALLOCATE RESOURCES
REPORT
COMPLETED
TASKS
PROJECT COMMITTEE
SPECIFY SKILLS
AUTHOR COURSES
EDITORIAL BOARD
REVIEW
MODIFY
APPROVE
REVIEW
BY
PROJECT
COMMITTEE
INSTRUCTIONAL PROGRAMMING
GROUP
CODE
TEST
DEBUG
RELEASE FOR REVIEW
BY PROJECT COMMITTEE
RELEASE FOR FIELD
TEST AFTER FINAL
REVIEW

Figure 1. Project Structure

and one representative from the OISE whose members review, modify, and eventually approve all work done by the Project Committee. The Instructional Programming Group of the OISE implements all approved materials on the computer facility.

In June, 1971, Cambrian College at North Bay, and Algonquin College in Ottawa joined the OISE and Seneca College in the project. At this time a one-month workshop was held for the teaching masters of the community colleges in computer-assisted instructional philosophy, methods and implementation techniques. Staff from Fanshawe and George Brown Colleges also attended the seminars. Currently, instructional materials are being authored by the aforementioned colleges with the addition of Sir Sandford Fleming and Conestoga Colleges. About three hundred students from three Toronto high schools are also currently using the curriculum. In addition to the preparatory mathematics curriculum which has been developed, curriculum for first year business and technology mathematics is under development.

Figure 2 shows the geographic distribution of the currently participating agencies which comprise our project's effort. As you can see from the figure, the project encompasses an area from North Bay to Guelph to Ottawa.

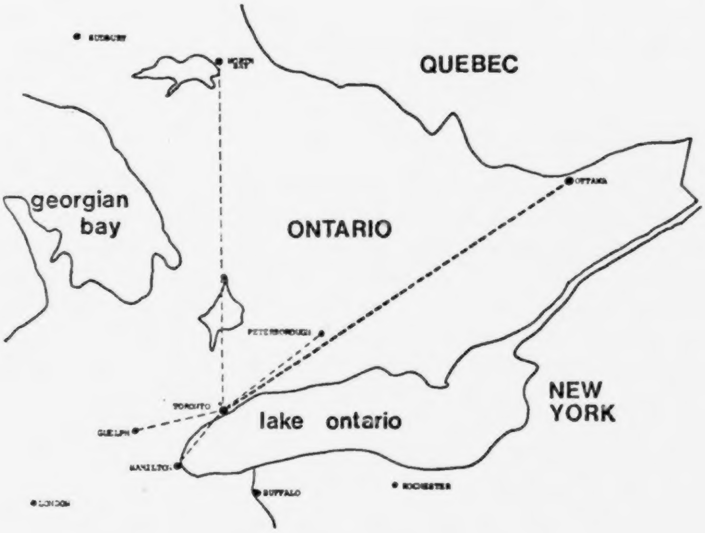

SUDBURY
NORTH BAY
QUEBEC
georgian bay
ONTARIO
OTTAWA
PETERBOROUGH
TORONTO
GUELPH
lake ontario
NEW YORK
HAMILTON
LONDON
BUFFALO
ROCHESTER

Figure 2. Location of Cooperating Agencies

Computer Facility

All of our users are connected over leased telephone lines to a small-scale, general-purpose timesharing computer which runs a line concentrator program as one of its higher priority jobs. The line concentrator at The OISE is connected to the National Research Council's (NRC) large-scale timesharing computer by a single, high-speed, leased telephone line (McLean, 1972). By the use of the line concentrator computer at The OISE, we may allow about thirty students on the system while using space for only one job on the PDP-10 at the NRC. Most of our courseware development is conducted on a medium-scale, general-purpose timesharing system at The OISE, and when the curriculum is debugged, the materials are transferred to NRC's machine for field trials. The student performance data recording and reporting facilities implemented on the PDP-10 allow us to make further refinements in the curriculum during the field trials.

Curriculum Materials Development

Let us focus our attention now on the curriculum development efforts. Teachers from the community colleges have specified the behavioral skills required for the target population students. The first task was to test the students to determine what instruction was needed. Rather than follow traditional testing methods, three computer dependent techniques were chosen.

Computer Generated Items

Instead of storing a large set of specific test items related to a required mathematical skill, only one model problem is stored. This model problem is parameterized so that it can produce a random sample of test items, drills or examples within the determined limits. There is one model problem for each skill and it is estimated that by using the generative technique for creating specific items the computer storage requirements have been reduced eighty percent over previous techniques. Of course, computer generation of items also relieves the instructor of this chore while expediting the task of coding and debugging items.

Sequential Testing

A second technique uses a sequential sampling procedure developed by Wald (1947) for industrial applications. The traditional decision procedure draws a sample of a predetermined size and computes statistics on this total sample. Wald's procedure samples items at random, until a dichotomous decision can be reached. This sequential procedure can reduce the required sample size by fifty percent. The implication is that the same information as to whether a student has a skill or not can be obtained with fewer test items and time. The way we have parameterized the Wald's Sequential Probability Ratio Test a student making frequent errors can take only two test questions, while a student making no errors may take at least four test questions (see Figure 3).

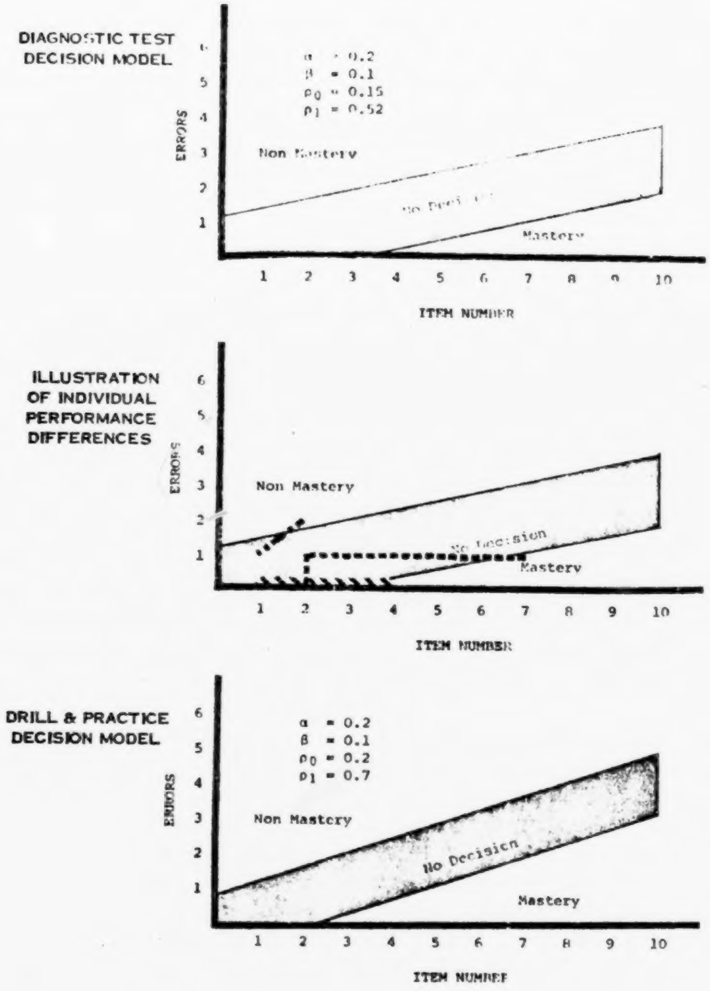


Figure 3. Sequential Probability Ratio Test

The vertical axis shows errors and the horizontal axis shows the total number of test items administered. The two sloping lines indicate the points above which a student demonstrates non-mastery on the test, and the area below which the student passes. The area between the lines defines the no decision region. As a student takes more test items, his point is displaced to the right. Any errors also cause an upwards displacement. Since it is possible to stay in the no decision region by making occasional errors, the testing procedure is truncated after a limited number of trials, and the decision region which will result in a minimal decision error is taken. These two facilities, generative techniques associated with the test items, and the sequential testing, would be difficult, if not impossible, without a computer. The only storage requirement is one program with approximately fifteen lines to control the sequential testing, and one of approximately twenty lines for each model problem.

Mathematics Skills Hierarchy

The third facet of the computer dependent technique is hierarchical branching. Most mathematics is considered hierarchical in nature; that is, certain skills are prerequisite to higher level skills. If a validated hierarchy is used as a testing structure then all skills need not be tested. For instance, if a student reliably can find the product of two factors each composed of two or more digits, then it may be inferred that the student has the skills of "multiplication facts" and "columnwise" addition. Such inferences, if valid, let one bypass testing lower prerequisite skills if the higher level skills are passed. By using the hierarchical structure, a great saving in testing time can result. The combination of the hierarchical branching and sequential testing techniques reduces the student's time involved in taking tests by 75% over the next best currently available testing techniques.

In order to create empirically and subsequently validate a hierarchy of relations between mathematics skills, it was first necessary to test a large population of students in all the prerequisite skills. OISE staff members created a "paper-and-pencil" test designed to simulate the computer's administration of diagnostic test materials. This test was administered at Seneca College, Kenner Collegiate in Peterboro and Northern Secondary School in Toronto. Information received from this first cycle of testing was sufficient to construct and implement the first test hierarchy in September, 1971.

The skills which are included in the prerequisite mathematics are shown in Table 1. For example, Figure 4 shows the skills specification for objective 6.1, which is one of the items of the complex fractions section. The student does not see the problem in the form shown in Figure 4, but rather, he sees specific values for the variables "a", "b", "c" and "d". The algebraic and set notation shown under the problem's format are the rules used in generating the problems. The sets for the variables "c" and "k" are numbers selected at random. The variables "a", "b" and "d" are thus transforms of the random variables "k" and "c".

Table 1

List of Mathematics Skills

1. Fundamental Operations with Signed Integers
2. Factoring of Integers
3. Laws of Exponents
4. Handling of Signs and Common Fractions
5. Equivalent Common Fractions and Reduction to Lowest Terms
6. Complex Fractions
7. Fundamentals of Decimal Notation
8. Operations with Decimal Numbers
9. Significant Digits
10. Algebraic Substitution
11. Addition of Monomials
12. Multiplication with Monomials and Polynomials
13. Division with Monomials and Polynomials
14. Monomial Common Factors

Divide a fraction by a fraction.

The question takes the form of the expression

$$(a/b)/(c/d),$$

where $a = kc$, $b = k + 1$, $d = k - 1$,
$c \in \{3, 5, 7, 11\}$, and $k \in \{4, 6, 8, 10\}$.

Figure 4. Skill Specification for Objective 6.1

Instructional Strategy

The instruction is typically a small amount of explanatory text followed by a few examples generated from the model problem for the skill. Following the examples the student takes a drill similar to the test, but now he is told when he is wrong. Comprehensive answer analysis in the model problems is invoked to see if his answer, if wrong, could have resulted from applying an anticipated incorrect procedure to the problem. If the student is making a common type of mistake he is made aware of this, and if it recurs he is taken to the objective lower in the hierarchy which teaches this skill. This is a review procedure for students who need it. While taking the drill the sequential pattern of the student's response is continuously analysed by Wald's procedure to see if a mastery or non-mastery decision can be reached.

Figure 5 shows the method of sequencing a student through the entire curriculum. This figure indicates that a student does not receive instruction on a skill until he masters the prerequisite skills. The first level of instruction for each skill is designed to pass about seventy percent of the students. More comprehensive instruction is given to the remaining thirty percent of the students who do not achieve mastery on the first level of instruction.

Evaluation

A preliminary evaluation of the project was conducted internally by Seneca staff. Table 2 shows the tabulation of these results. These results, of course, should not be taken as definitive, nor in any sense do they "prove" the benefits of this type of approach. The results from Seneca should be taken only as a preliminary indication. The major findings of Seneca's internal evaluation are the lower dropout rates, near absence of failures and drastically reduced time for the CAI group.

Currently, The OISE is collecting performance data from all of the users and effecting a large scale evaluation study.

Conclusion

I have attempted to show how the OISE and several Colleges of Applied Arts and Technology jointly are developing a curriculum for upgrading mathematical skills. The project uses the CAI facilities of The OISE and the NRC to administer generative diagnosis and instruction for mathematical skills. The mathematics curriculum material is designed as a set of hierarchical modules keyed to specific behavioral objectives prerequisite for first year community

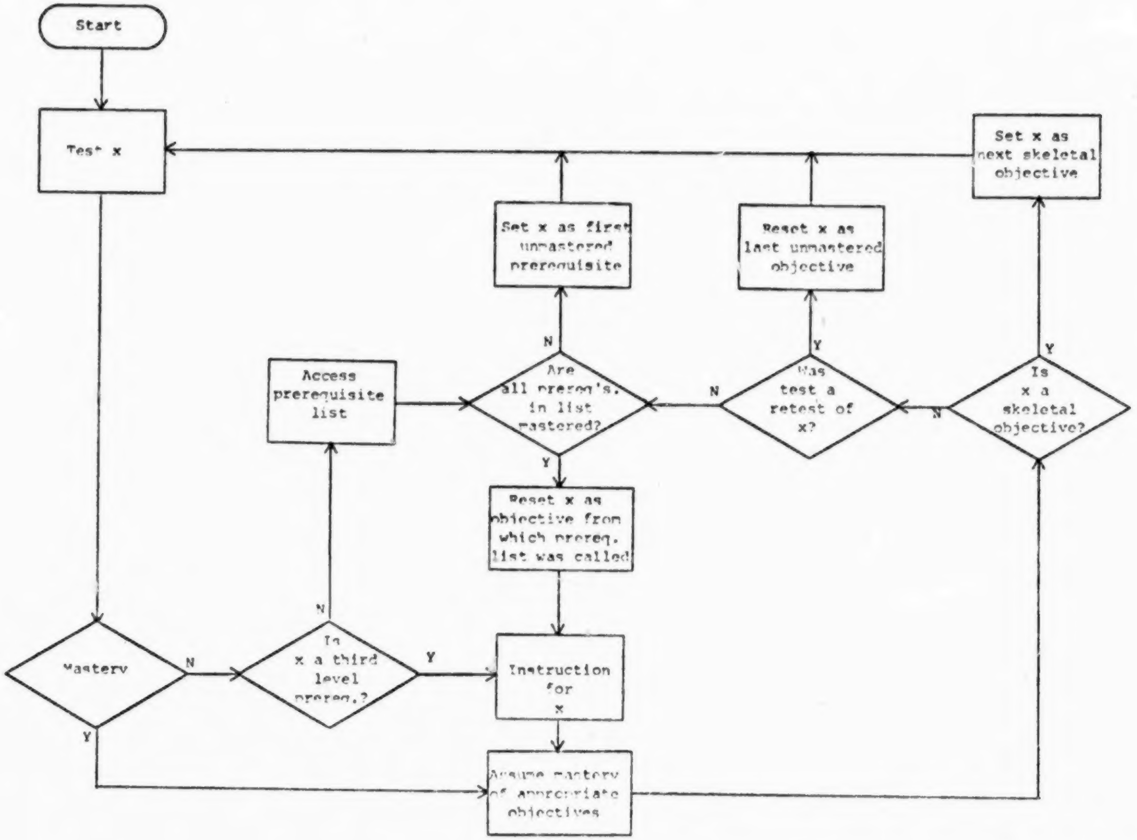

Start
Test x
Set x as next skeletal objective
Set x as first unmastered prerequisite
Reset x as last unmastered objective
N
Y
Access prerequisite list
Are all prereq's. in list mastered?
N
Was test a retest of x?
N
Y
Is x a skeletal objective?
Y
Reset x as objective from which prereq. list was called
N
Mastery
N
Is x a third level prereq.?
Y
Instruction for x
Y
Assume mastery of appropriate objectives

Figure 5. Sequence of Progression Through Hierarchy of Objectives

college students.

The goal of the project is to provide the student with the most efficient means of instruction suited to his individual needs. The advanced assessment techniques requiring an on line computer system provide valid decisions which can reduce by seventy percent the student's time required for mastery of the curriculum. The project is nearing the end of the first year of field trials and soon will produce formal evaluative results; however, preliminary evaluation studies have indicated an enormous utility for this type of approach.

Table 2

Evaluation Study by
Seneca College of Applied Arts and Technology

	Group		
	I	II	III
Nature of Instruction	Conventional	Conventional	CAI
Date	Summer '71	Fall '71	Fall '71
Student contact hours	28	40	6-28
Number of students registered	40	41	27
Number of students who transferred to another class	0	7	1
Number who passed	24 (60%)	7 (17%)	16 (59%)
Number who failed	8	10	9 incomplete*
Number who dropped out	8	17	1

*These students did not complete the material and continued into the spring of 1972.

Bibliography

Churchill, Stacy, Naess, Lars & Olivier, William P. CAN-4, An Advanced Author Language for CAI, Computer-Based Testing and Psychological Experimentation: PDP-9 Implementation. *DECUS Proceedings Spring 1970.*

McLean, Robert S. *A Model of a Centralized CAI System.* Paper presented to Canadian Symposium on Instructional Technology, University of Calgary, Calgary, Alberta, May 24-26, 1972.

Wald, Abraham. *Sequential Analysis.* New York: John Wiley & Sons, 1947.

SIMULATING AN INSTRUCTIONAL SYSTEM FOR AN EDUCATIONAL GAME

P. David Mitchell

Director
Graduate Programme in Educational Technology
Sir George Williams University
Montreal

SIMULATING AN INSTRUCTIONAL SYSTEM FOR AN EDUCATIONAL GAME

SESSION 6A
ROOM/SALLE 145

By/Par: P. David Mitchell,
Sir George Williams University.
Université Sir George Williams.

Research and development activities to be described emerged from a twofold problem: How can the responsibility for learning be shifted from teachers to learners so the learner allocates his time or other resources to achieve preferred educational objectives and the teacher develops and evaluates strategies to assist the learner? How can the teacher-trainee be prepared for this role reversal?

A computer simulation of an instructional system (i.e. a set of learners) is being developed. The game-player, cast in the role of a teacher, takes pre-instructional decisions. The model calculates coefficients for each decision, updating major sub-models within the simulation (motivation, instruction, information and activity states) for each learner in the simulated population with regard to several educational objectives. It prints results of educational assessments called for by the game-player.

The game-player is able to test the effects of alternative instructional strategies and supportive action on the simulated students. The player may resort to lengthy trial-and-error, retrieval of information from various resources, discussion or higher cognitive skills. The onus is placed on the player to find one or more winning strategies. This is in keeping with the model's assumption that the learner is an adaptive, self-organizing system which begins in a purposeful state and responds differentially to instructional communications and activities. Thus the game-player is forced to adopt a learner-centered approach both experientally and in any winning strategies.

When the educational game is operational it will be tested with teacher-trainees, experienced teachers and others. Improved models of the simulated classroom are destined to become a new focal point for teacher training activities.

LA SIMULATION D'UN MODÈLE D'INSTRUCTION D'UN GROUPE D'ÉLÈVES COMME JEU ÉDUCATIF POUR LES ENSEIGNANTS

Les travaux de recherche et de développement que nous décrivons sont nés d'un double problème: comment peut-on donner la responsabilité d'apprendre à l'élève plutôt qu'au professeur de sorte que l'élève consacre son temps ou d'autres ressources pour atteindre des buts éducatifs choisis et que le professeur développe et évalue des moyens d'aider l'élève? Comment peut-on préparer le professeur stagiaire à ce rôle?

On développe une simulation sur ordinateurs d'un système pédagogique (c'est-à-dire, d'un groupe d'élèves). Le meneur de jeux, le professeur, prend des décisions d'instruction anticipées. Le modèle calcule des coefficients pour chaque décision, mettant à jour les sous-modèles majeurs à l'intérieur de la simulation (la motivation, l'instruction, les renseignements et les états d'activités) pour chaque élève dans la population simulée à l'égard de plusieurs objectifs éducatifs. Il imprime les résultats des évaluations éducatives demandés par les joueurs.

Le joueur a la possibilité d'essayer les effets d'autres stratégies éducatives sur les élèves simulés. Le joueur peut aussi faire des essais et des vérifications de longue durée, récupérer les renseignements de différentes sources, discuter ou employer des techniques impliquant des connaissances supérieures. C'est au joueur de trouver une ou plusieurs stratégies gagnantes. Ceci s'accorde avec l'hypothèse du modèle que l'élève est un système adaptif à organisation propre, qui commence avec un but avisé et qui répond, en différenciant, aux communications et aux activités éducatives. Ainsi le joueur doit adopter une attitude d'élève désirant apprendre aux moyens des expériences et des stratégies gagnantes.

Quand le jeu éducatif sera mis au point, on fera des essais avec des professeurs stagiaires, des professeurs expérimentés et d'autres. Des modèles perfectionnés des salles de classe simulées auront une place très importante dans les études des stagiaires de l'enseignement.

At Sir George Williams University we are currently attempting to develop an educational game in which the game player, cast in the role of teacher, will take pre-instructional decisions and a computer program will simulate an instructional system and provide plausible instructional outcomes for a set of students. This paper provides an overview of the educational simulation (EDSIM) project.

THE MANAGEMENT OF LEARNING

Recent educational technology research and practice has emphasized not only the institutionalized replacement of some instructional functions of the teacher by other means and modes of instruction but also the changing nature of the teacher's role toward that of manager of the educational process. Somewhat disquieting is the realization that little has changed in teacher-training practices to implement this.

The typical teacher-training program focuses on knowledge and understanding of the discipline as well as on teaching methods. But it gives only lip service to the transformation of discipline content and psycho-social sciences into pedagogical practice through mediated instruction and learning systems. The teacher is left to his own devices while pursuing the elusive state of omnicapability. Yet the classroom teacher cannot be expected to develop the kinds of learning resources and environments likely to prove necessary; it must be a large-scale effort to be economically feasible. Admittedly a teacher may prepare some learning and instructional resources but not in sufficient quantity and quality to make effective use of his time and talents. One promising development is the trend to multi-purpose classrooms characterized by small groups of students working on defferent projects. Another is that learning resources are gradually becoming available in schools and universities. However there is still a problem; how can we help the beginning teacher learn to design and manage the conditions of learning for his students?

A closely related problem exists. Emphasis on technological efficiency in instructional communications could destroy efforts to develop self-regulated learning strategies on the part of students. For the basic issue is not the improvement of teaching but the improvement of learning. This is a subtle point based on the assumption that the activities and attitudes of the student are necessary (and often sufficient) whereas those of the classroom teacher or instructional system may be necessary but are certainly not sufficient. Ultimately the student is his own teacher; he must come to see that he is responsible for his own education—and that his experiences and activities produce changes in his capability. Probably most students would fail if they did not teach themselves outside, and even inside, the classroom. So the student, as much as the teacher, is the manager of learning; they must collaborate.

THE PROBLEM

The problem is twofold: How can the responsibility for learning be shifted from the teacher to the student so the latter can allocate his time or other resources to achieve his assigned or preferred educational objectives while the teacher functions as a manager of learning resources for his students? How can the student of education be prepared for this role?

A SIMULATED CLASSROOM

Although other possibilities exist, current research is based on the hypothesis that simulation and gaming methods which have emerged in military and business settings may provide a useful approach to this problem. The simulated environment provided by a game usually is less complex, and therefore more manageable, than the actual situation. It has pedagogical advantages such as more rapid knowledge of results, opportunities to try various strategies without intervention in a real system,or the possibility of testing theoretical propositions more readily. Further, an educational game can be designed so that other resources, such as books, media, other game players or faculty consultants, are required for a meaningful experience on the part of the player.

AN OVERVIEW OF THE EDSIM PROJECT

There exists a plethora of descriptive and explanatory literature concerned with education, instruction and learning. In spite of this, slow progress has been made toward a viable theory of instruction, much less a generally applicable quantitative model. Yet a quantitative model of the instructional process--no matter how crude--is required if a computer simulation is attempted. For such simulation is a "numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behavior of an .../organized/ system (or some component thereof) over extended periods of real time" (Naylor, Balintfy, Burdick & Chu, 1966, p. 3). A more precise definition is ""x simulates y" is true if and only if (a) x and y are formal systems, (b) y is taken to be the real system, (c) x is taken to be an approximation to the real system, and (d) the rules of validity in x are non-error-free" (Churchman, 1963, p. 12).

One variant of simulation is operational gaming, in which a human game-player participates as a decision-taker within the structure of the system being simulated. EDSIM will be such an educational game.

OBJECTIVES

Four inter-related broad objectives which will require considerable research extending over a two or three year period have been delineated:

1) We will develop a simple and crude conceptual model of a learner as an adaptive, self-organizing system which begins in a purposeful state and responds differentially to educational communications designed to motivate, inform or instruct this system. This model will then be refined in the light of psychological and other information (including teachers' opinions) and transformed into a computer-based model of a learner which approximates the kinds of behaviour and decisions likely to be observed in a realistic, albeit restricted, instructional system.

2) After the model of a learner has been developed it will be used as a simulated student upon which various instructional communications strategies can be tested for their effects. This "standard learner" will serve as an experimental subject for students of education who will attempt to induce a positive increment in the model's capability state (i.e. the learner's knowledge, attitudes, or behaviour) with respect to specified objectives. As a result of these trials the model doubtless will have to be revised.

3) Several standard learners can be generated from the basic model. They may be duplicates of the original or can be made to differ simply by changing initial values of the purposeful state. The set of learners, which makes up a simulated classroom, will form the core of the educational game in which the game-player, cast in the role of teacher, takes pre-learning decisions regarding selection of media, modes, or content of instructional communications or alternative educational activities. The simulated instructional system will update the capability state both for each learner and for the class as a whole with regard to several learning or activity objectives.

4) The simulated instructional system will be used as a learning aid for students of education. By encouraging them to utilize a variety of learning resources in developing a winning strategy for the game (which itself is learner-centred), the game may help teacher trainees to adopt a learner-centred approach to education. More importantly, it will provide practice in thinking precisely about learning needs and it should promote flexibility in managing learning resources to meet such needs.

THE EDSIM MODEL

Presently sought is a flexible simulation model which permits the user to specify the number of students to be included in the instructional system (to a maximum of thirty) and initial characteristics of each (e.g. slow learner, under-motivated, highly motivated). At present, however, no attempt is foreseen to permit the user to generate alternative models of human learning; the "standard learner" will remain the core of the model. Ultimately such flexibility may seem desirable.

The EDSIM is thus to be a simulation game involving a hypothetical class, which is simulated by the computer program, and a teacher or game player. The model is based on assumptions that are not entirely realistic, in the interest of simplicity. For instance, although some flexibility exists in stipulating student characteristics, only one model of learning style is planned. Therefore all students are essentially identical even though they will not appear so throughout the game. More realistic and complex assumptions can be accommodated at a later date if desired.

In real life, the teacher may play many roles in fulfilling his responsibility to the student and society. Yet the EDSIM model permits the game player only a limited number of alternative activities to bring about changes in the capability state of simulated students. No attempt will be made to include classroom behaviour problems; the aim is to focus on pre-instructional decisions (e.g. lesson planning) of the teacher and their probable effects on a simulated classroom. In practice, other methods of teacher training such as interaction analysis, in-basket studies, micro-teaching or practice teaching may be expected to supplement EDSIM.

Another limitation, the curriculum, may be turned to advantage. The present intention is to limit the curriculum to a set of ten mathemata (from Greek; mathémata = things learned). The mathemata are to be structurally organized so that some mathemata are more easily learned if certain of the others have been learned. In such a hierarchical arrangement (cf. Gagné, 1965; 1968), if mathemata subordinate to higher-order mathemata have been learned, there is a high probability that the student can display the higher order capability (assuming adequate instructional communications). On the other hand, the probability is low if the prerequisite capability is missing. This does not rule out learning in the latter case; a finitely large number of instructional acts may result in the learning of both higher- and lower- level mathemata. In the present model, this practice, while possible, is extremely inefficient in use of time. Because the game player will be given only a verbal description of the ten mathemata in the curriculum, not a chart, he must implicitly, or perhaps explicitly, devise his own teaching strategy for the mini-curriculum. Appropriate references and resource material on curriculum development will be provided for his use. Thus the curriculum is not limited to any discipline but should be easily interpolated by the game player to fit any convenient subject. However, it will be limited to the cognitive domain of educational objectives and the lowest levels of the affective domain.

To begin playing the game the player is required to take several decisions, for example: what is to be taught in the forthcoming period, whether he will teach the entire class, a designated group, or only one student; whether to assign homework, give a test, or teach a learning strategy;

whether to induce recall of relevant mathemata, etc.. Essentially the model allows only one topic to be taught at a time but more than one may be taught in a period. A set of pre-instructional decisions for a period is read into a computer that has been programmed with models that alter the capability state of the students as a function of the game player's decisions. In addition, chance plays a role in determining the learning outcome.

In brief, the model calculates coefficients for each decision on the part of the game player. These coefficients will be used to modify one or more of the major sub-models within the game (viz. motivation, instruction, information, activity, and capability states). The importance of the motivation sub-model in this game cannot be overemphasized. The model assumes a general motivation to engage in learning, within or beyond the classroom, and a motivation to undertake specific learning tasks directed toward specified outcomes as might be determined by the teacher or the student. Motivation state could be altered by appropriate instructional communications and will be enhanced if the student is taught learning strategies independent of the curriculum.

Similarly other sub-models would mimic what is inutitively obvious and relevant in learning. Information state would be a function of motivation and instruction as well as of the number of students involved in instruction in a given period. Test performance is closely related to the student's capability state but may vary, depending on circumstances at the time of testing. And capability itself may be dynamic and directly observable or latent (i.e. presumed to exist but not demonstrated; cf. Mitchell, 1972).

SUCCESSFUL GAME PLAYING

The unit of play is a simulated fifty minute operating period. Since each teaching act uses up an arbitrarily assigned number of minutes, only a limited number of teaching acts may occur in any period; the game player is thus encouraged to be economical in his use of time.

The player of the EDSIM game will be able to test the effects of alternative instructional and learning strategies on the simulated students. The player may resort to lengthy trial-and error retrieval of information from various resources (to be provided), discussion with teachers, students, or fellow trainees. Or he may rely on his own cognitive strategies for problem solving. The onus is placed on the player to find one or more winning strategies. This is in keeping with the model's assumption that the learner is an adaptive, self-organizing system which begins in a purposeful state and responds differentially to instructional communications and educational activities. The game player, then, is forced to adopt a learner-centred approach both experientially and in any winning strategies.

The game might be organized in either of two ways. First there can be a time limit placed on teaching whereby the 'course' terminates at the end of a specified number of instructional periods. Teaching effectiveness of the player could be determined by an assessment of what the class has learned. The second mode involves running the game until a specified number of students has reached a predetermined criterion. Thus time would be treated as a variable and degree of learning as a constant; teaching success is then a function of time required for the player to achieve this criterion.

In either of the illustrated playing modes, the player can compete against his own past performance or with other players, or he may be required to achieve a predetermined criterion.

When the educational game is operational, we plan to test it with a suitable class of education students and assess its effectiveness in combination with traditional approaches in training education students to apply their academic courses to instructional problems. Appropriate learning resources will be selected or produced to assist the game player who wishes to improve his performance. The game players' reactions will be sought. Improved models of the simulated instructional system seem destined to become a focal point for teacher training activities.

REFERENCES

Churchman, C.W. An analysis of the concept of simulation. In Symposium on simulation models. Cincinatti, Ohio: South-Western Publishing Co., 1963.

Gagné, R.M. The conditions of learning. New York, N.Y.: Holt, Rinehart and Winston, Inc., 1965.

Gagné, R.M. Learning hierarchies. Educational Psychologist, 1968, 6, 1, p.1 et passim.

Mitchell, P.D. The sacramental nature of behavioural objectives. Paper presented to the International Conference on Educational Technology at Bath, England, March 1972. In Aspects of Educational Technology. Vol. 6 (in press).

Naylor, T.H., J.L. Balintfy, D.S. Burdick & K. Chu. Computer simulation techniques. N.Y.:Wiley, 1966.

ACKNOWLEDGEMENT: This research is supported by a grant under the Québec Minister of Education's programme of "Formation de chercheurs et d'action concertée."

THE USE OF A TERMINAL BASED SIMULATION IN MANAGEMENT EDUCATION

By

Sandy Douglas
and
John R. Abrahams

-- Ryerson Systems Institute --

SESSION 6B
ROOM/SALLE 145

THE USE OF A TERMINAL BASED SIMULATION IN MANAGEMENT EDUCATION

By/Par: John R. Abrahams, Sandy Douglas
Ryerson Systems Institute

The education of computer user management is being recognized by an increasing number of organizations as a good method to improve communications and the effectivity of computer services. One way it can do this is by removing the sometimes deeply engrained distrust of the computer and by showing its real capabilities and limitations. With a more positive attitude, the user manager is able to look around his own department, technical specialty or profession and identify some possible opportunities for computerization. He will also be more likely to interact successfully with the Systems Department on systems projects which affect his area.

A terminal based simulation has been found to be one of the most effective methods of removing the fear of the computer and showing its capabilities. In the first place, the simplicity of the terminal permits students to grasp the function of the computer without being overawed by the complexity of the hardware. Secondly, they are put in a position where they actually control the computer through the terminal and use it as a tool. Thirdly, it is a good springboard for a relevant discussion of many aspects of computing and particularly time-sharing, telecommunications and simulation. Finally, the terminal based simulation will bring the class together in working groups and effectively initiate participation and dialogue.

Over the past 18 months, Ryerson Systems Institute has written 3 such simulations of increasing complexity and used them extensively in both user management education and in Systems Analyst education.

L'UTILISATION D'UN TERMINAL EN SIMULATION UTILISÉE POUR FORMER LE PERSONNEL DE GESTION

Des organismes, de plus en plus nombreux, reconnaissent que les ordinateurs sont utiles pour former le personnel de gestion, pour améliorer les relations entre humains et le rendement des ordinateurs. L'une des façons dont ils peuvent le faire est de dissiper la méfiance, souvent profonde, à l'égard de l'ordinateur et de montrer ses applications éventuelles et ses limites. Si le gérant qui s'en sert a une attitude favorable, il peut, en cherchant dans son département, dans sa spécialité technique ou dans sa profession, trouver des possibilités de se servir de l'ordinateur. Il aura probablement aussi plus de succès dans ses relations avec le Département des systèmes au sujet de ses travaux.

La simulation à l'aide d'un terminal apparait comme l'une des meilleures méthodes pour surmonter la peur éprouvée en face de l'ordinateur et de montrer ses possibilités. Tout d'abord, la simplicité du terminal permet aux étudiants de comprendre les fonctions de l'ordinateur sans être intimidés par sa complexité. D'autre part, ils peuvent eux-mêmes contrôler l'ordinateur par l'intermédiaire du terminal et l'utiliser comme outil. Une troisième raison se trouve dans le fait que le terminal est un bon moyen de déclencher une discussion pertinente sur de nombreux aspects de l'emploi de l'ordinateur, soit à temps partagé, soit pour la simulation ou les télécommunications. Enfin, la simulation à l'aide de terminaux réunira les élèves de la classe sous forme de groupes de travail et elle rendra efficace la participation et le dialogue.

Durant les dix-huit derniers mois, le Ryerson Systems Institute a programmé trois simulations dans ce cadre, de difficulté croissante et il en a fait un usage considérable dans la formation de gérants et d'analyseurs de systèmes.

The Use of a Terminal Based Simulation In Management Education

Providing education in the use of the computer to diverse groups of management and computer professionals creates some interesting problems. At the Ryerson Systems Institute, we have found computer based simulations to be extremely valuable tools for solving some of these problems. Over the past two years, we have written three such simulations and used them in a wide range of courses.

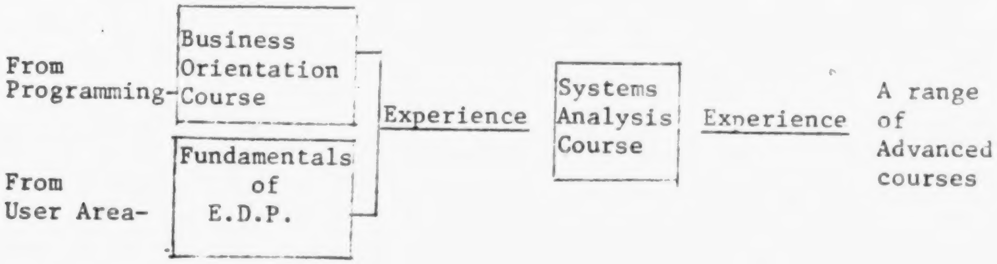
Ryerson Systems Institute was founded in April 1970 and is part of Ryerson Polytechnical Institute in Toronto. The aim of the Systems Institute is to provide education in the area of information processing to both managerial and technical staff in government and business. In order to do its job more effectively, it carries out applied research in information processing education and provides a centre in which information processing professionals may share ideas. Though a part of Ryerson Polytechnical Institute, the Systems Institute is financially self-sufficient and partially supported by a consortium of major Canadian companies. It is governed by Ryerson, representatives from the consortium members and representatives of each of the three data processing, professional associations. The Institute, in cooperation with the Data Processing Management Association, has instituted a comprehensive series of evening courses and these will be made available to other educational institutions in the near future. In addition, in-house or specifically tailored courses have been prepared and presented for numerous organizations including Imperial Oil, Simpsons-Sears, Shell, The Canadian Institute of Chartered Accountants, Royal Bank, Domtar, C.I.L., Public Service Commission, Canadian Armed Forces, Toronto-Dominion Bank and C.N.A. Assurance.

Courses are designed for two rather dissimilar groups of people. The first group contains manager-users who hope to gain an understanding of the computer and business systems generally. This group does not come to learn specific skills but rather is eager to gain a general understanding of the area. The company which sends them is hoping that they come back with a positive attitude towards computer systems and the ability to interact successfully with the systems department. Numerous, well publicized computer system failures and a history of the computer's doubtful contribution towards profit in many organizations ensures that many of these managers come to the course deeply skeptical. Others, fear that the computer will mean the technological obsolescence of their job and they come very guarded or even bitter. The great problem, therefore, is how can we affect a change of attitude. One, very effective answer is to break down the barrier between man and computer by allowing him to "play" with it, to use it, to manipulate it and to control it. A simulation allows this.

How does a simulation encourage users to work with and control the computer? It does so by allowing him to stay on his own ground while the computer handles the unfamiliar aspects of the situation. The manager/user most often need do no more than manage effectively in using the simulation. Let us take an example. A group of retail store buyers have come to a course and for the reasons mentioned may be a little skeptical or dubious about the computer. After a short introduction, they are divided into management teams and each given a booklet describing the computer-based simulation with which they will be dealing. Although they are somewhat doubtful at first, they realize from reading the booklet, that they are on familiar ground. The simulation is of a retail market and all they need do is make the merchandising decisions, with which they are completely familiar. The simulation program will act as the market and show each team the effect of their decisions at regular intervals. Teams set prices, order merchandise and establish stock levels; then the simulation program produces the resultant sales for each item.

The most important aspect of this approach is that it initially portrays the computer as a valuable tool rather than as a mystifying black box. The approach emphasizes useability. From this point the discussion of almost all other areas of computing can be approached without, or at least with less, problem from attitude barriers. What a program is and how it is used can be illustrated using the simulation program. The principle of Input - Processing - Output becomes evident as the participants use the computer. Depending on which is being used, either the minicomputer or time-sharing can be effectively demonstrated and the team members given an opportunity for hands on experience. The question of what is simulation and how can it be applied to business situations can be answered and participants will be encouraged to consider on the job situations where it might be applied. Finally, it can act as a springboard for a discussion of systems and computing in their own sphere of influence. Could the output be more effective, more detailed, more specific or better organized? Could the processing be done differently and is the data captured in the system adequate in terms of the output and objectives?

The second group of people which our courses address is the information processing professional; most commonly these are individuals aspiring to become Systems Analysts. A Systems Analyst is the person who researches computer user's needs and translates them into a computer system. These people come from two, quite different backgrounds in almost equal numbers. The first background is computer programming and in some cases, computer operations. They usually have an adequate understanding of computer equipment but rather little appreciation of business organization and information flow in business. The second background is a user area. These people usually have a good understanding of the organization, the needs of the user area, the flow of information and any organizational problems which might be encountered, but they have inadequate equipment knowledge. The approach, therefore, in providing systems analyst education is to provide exposure to the areas that are not part of an individual's background. Handling these groups is further complicated by the fact that classes usually contain representatives from both areas.



Once again, a simulation in which participants work in teams is one effective tool for dealing with the problem. A properly designed simulation will identify characteristics and problems of the business environment as well as illustrating characteristics of the technology. Participants who have a data processing background can get a feel for the information requirements of a user area and some of the problems of management. Without realizing it, they may begin to think like users and that can have a very beneficial effect on their ability to communicate with user groups. Participants with a user background will, of course, benefit from exposure to a computer and to simulation in much the same way that the manager/user groups did. An interesting side effect that comes when using a simulation in a group with different backgrounds is the way which it will get them working together. It is certainly worthwhile having computer specialist and user specialist working together as this cooperation is critical for effective systems work. Like so many actual instances, a simulation can make these two groups collaborate on a computer based problem, which paradoxically may be solved most effectively through the user's contribution.

As an example, a group of employees is on a course from a large bank. A portion of this group has come from the data processing area and the remainder from various user areas, such as branches, accounting, the mortgage department and others. They are divided into teams of four or five and given a booklet describing a simulation. In order that neither group has a marked advantage, the simulation deals with an unrelated business. It is more elaborate than the first one we mentioned and requires more complex decision making and constant communication between team members. The teams in this case must run a manufacturing company by making decisions on production and stock levels and bidding for orders from customers. The computer reports their performance in a statement at the end of each simulation week and poses other problems like fires, strikes and equipment breakdowns which the teams must deal with. To be effective, each must pick a specialty such as General Manager, Sales, or Accounting, they must devise an organization structure and a system of communications.

This immersion in a simulated business situation has been shown to help the potential systems analyst think like and relate to management. Secondly, he does gain a better understanding of information flows and requirements in user areas. Finally, it offers the problems of effective team work, which many individuals, particularly programmers who are accustomed to working alone, may not have experienced.

The Systems Institute has made an attempt to satisfy it's requirements by writing three simulations. Others will be written as required for specific courses or clients. Here is a brief description of the three.

Searton Fashions is intended for use in computer appreciation or manager/user courses. It is the shortest in terms of class time, taking only two and one-half hours and is the least complicated for the participants. The recommended team size in this case is three to six persons and four teams is considered a maximum. It was originally written for a high school course, known in Ontario as "Information Processing Concepts", but has been used with a number of groups since then. Interestingly enough, it has been used extensively with buyers from Simpsons-Sears Limited who have definitely enjoyed it and provided valuable feedback which we have incorporated into the simulation. Searton is most effective when run manually for three simulation weeks and then through the terminal for seven.

This approach demonstrates, very graphically, the computer's ability to remove onerous clerical chores and allow time for effective decision making.

Computer Utility Management Corporation is a simulation which can be used in either type of course. In this simulation, the participants manage a company which sells computer time. The teams must set prices, plan advertising and service standards for each simulated quarter and the computer reports their performance. It takes approximately three hours of class time and will accomodate up to four teams of two or three persons. It is a good vehicle for introducing the basic ideas of time sharing as in effect "the medium is the message". It highlights the financial aspects of such a system and identifies some of the technical problems which may be inherent. Although it was originally used in "computer appreciation" courses, Computing Utility has been purchased by several large organization, such as the Public Service Commission, for use in Systems Analyst training. It is often used in answering the question "What is and how can we use a simulation program?"

Plastics Incorporated is a general-purpose business simulation which dramatizes the communications problems and information requirements within a management team. It is used primarily in systems analyst or management education, although it has been employed in more elaborate manager/user courses. In this simulation, the teams must set up a management team with specific job assignments and a definite communication system. They will then act as a management team in the environment simulated by the computer. Many decisions have to be made each simulation week and some interesting exception conditions have to be dealt with. Obviously, the success of the team depends to a large extent on its ability to organize, define authority and communicate. This is the largest of the current simulations and requires from six to fifteen class hours. It can accomodate up to four teams which consist of a General Manager, an Accountant, a Sales Manager, a Production Manager and a Systems Analyst. In addition to its use in Systems Analyst training, it is used in the final year of an Industrial Engineering degree and in several systems management courses.

Now for a few of the technical details of these simulations. All of them were originally written for use on time sharing services, such as IBM/CALL 360. This arrangement was advantageous since it made the programs available across the continent and a number of our larger clients already had this service. They can also be run on a variety of mini-and full size computers. All of the programs are written in BASIC because of its capability and extensive availability on both mini-computers and time sharing services. A variety of terminals can be used, varying from a teletype, through IBM 2741 to video terminals such as the Lektromedia Unit. Total running costs on a time sharing service are around $10 per hour. The simulation program sizes are as follows:

Searton Fashions	-	10k bytes
Computer Utility	-	9k bytes plus one file
Plastics Incorporated	-	23k bytes plus one file

These simulations in the form of program listing and paper tapes are available for purchase.

The ratio of design and test time to classroom time varies with the size of the simulation. The shortest simulation had a ratio of about 60 to 1, runs for about three hours in the classroom and therefore the development time was about 180 hours. At $10 per man hour, the simulation cost some $2,000 to develop from scratch. The longer simulation had a total development ratio of about 200 to 1.

Each of the programs can handle up to six teams and four teams seems to be the optimum number. With four teams, each team member spends two or three minutes per hour, actually using the terminal. We have also found that the simulation should normally occupy about 20% of total course time.

You will have noticed that this approach is not one-for-one C.A.L. Neither, however, is it a batch processed business game. The simulations can accomodate from six to sixteen persons per terminal and if necessary more than one terminal can be used. This makes it possible to incorporate the simulations into courses easily since the whole group can be involved. The team work stimulated by this approach is also an important aspect of a number of the courses in which the simulations are used. Unlike a batch processed business game, these do give the teams constant feedback on their progress. They also force the participants to plan at the same time as they handle the constant crises or exception situations that typify business. In addition, the simulations permit one instructor to control a case study situation more effectively than in the past. The Plastics Incorporated simulation, when it was done manually, required one frantic instructor for each of the four teams. One instructor can now easily control the entire simulation.

In conclusion, we, and others who have used these simulations have found them to be dramatic teaching vehicle suitable for a wide variety of instructional situations.

TEACHING LOGIC WITH CAL

USING APL

David Vaskevitch

SESSION 6C
ROOM/SALLE 145

TEACHING PROPOSITIONAL CALCULUS WITH C.A.L. USING APL

By/Par: D. Vaskevitch,
Erindale College,
University of Toronto.

This paper describes a series of lessons designed to teach Propositional Calculus which can be used by anyone from school children to university graduates. The lessons are written in the form of games based in many respects on a commercially available kit called "WFF'N'PROOF" which has been very successful in motivating students to learn symbolic logic.

In most CAL lessons, the dialogue between the student and the computer is rather one-sided: the computer asks a question and then alters the presentation of the lesson based upon the student's response. However, the student has no way of asking questions of the computer.

In these lessons, the student is given many opportunities and even required to ask the computer questions. These questions, which take the form of theorems to be proved, are set up in the highly formalized notation of the propositional calculus. The computer responds by actually constructing a detailed proof of the theorem and if necessary leads the student through the proof line by line. This paper explains how the lessons are written and includes a discussion of the theorem prover used to construct the proofs using the method of subordinate proofs. The use and implementation of these games and lessons at Erindale College are discussed. While there are not yet sufficient lessons to teach all the rules of inference, those already available have been very successful.

L'ENSEIGNEMENT DU CALCUL PROPOSITIONNEL À L'AIDE D'ORDINATEURS ET DE L'APL

Nous décrivons une série de leçons conçues pour enseigner le calcul propositionnel à l'usage de tout élève, depuis les élèves des écoles jusqu'aux diplômés universitaires. Les leçons prennent la forme de jeux basés, sous beaucoup de rapports, sur un jeu que l'on peut trouver dans le commerce appelé "WFF'N'PROOF" qui a eu beaucoup de succès pour encourager les élèves à apprendre la logique symbolique. Dans la plupart des leçons EAO, le dialogue entre l'étudiant et l'ordinateur est plutôt unilatéral: l'ordinateur pose la question, puis il change la présentation de la leçon selon la réponse de l'élève. Par contre, l'élève n'a pas la possibilité de poser des questions à l'ordinateur.

Dans les leçons que nous proposons, l'élève peut et il doit même, poser beaucoup de questions à l'ordinateur. Ces questions, qui sont sous la forme de théorèmes à prouver, sont posées sous la forme de la notation très formelle du calcul propositionnel. L'ordinateur répond en faisant une démonstration détaillée du théorème et, si nécessaire, il entraîne l'élève, pas à pas, jusqu'à la fin de la démonstration. Nous expliquons comment on écrit les leçons et nous discutons de la méthode de démonstration basée sur des démonstrations subordonnées. Nous discutons de l'utilisation et de la mise en oeuvre de ces jeux et de ces leçons au Collège Erindale. Quoiqu'il n'y ait pas encore assez de leçons pour enseigner toutes les règles de déduction, celles qui existent ont eu beaucoup de succès.

Copyright: WFF'N'Proof - Learning Games Associates
1111 Maple Avenue, Turtle Creek, Pa. 15145 U.S.A.

TEACHING LOGIC

PREFACE:

This paper describes a project currently under active development at Erindale College. The purpose of the project is to design and implement a package, in the APL language, suitable for teaching elementary symbolic logic, at the undergraduate level, using Computer Assisted Learning (CAL).

The project, as it is presently constituted, consists of the work of the author, as described herein.

When completed the package will allow students to learn all the elements of logic at a computer terminal. This will allow teachers to leave the learning of the formal manipulations involved in the proof procedure to be done outside of lectures.

As this paper is being written the project described approaches completion and it is anticipated that the first classes to use it will do so in the academic year of 1972-73.

TEACHING LOGIC

INTRODUCTION:

It has long been the belief of the author of this paper that CAI is headed away from the type of conventional lesson produced in the last decade. On e of the chief characteristics of this type of material has been that the computer has largely played a "question asking" role. In this role it has not been called on to use much of its problem solving abilities.

Typically, instead, it has asked questions, and based on the results of analysing the student's answer has decided what further questions to ask. The only place where it has to use its capabilities at all is in analysing the student's answer. While this is an important ability to be have (i.e. response analysis) typically it does not demand very much of the computer. In fact it cannot demand very much of the computer because the analysis must be very rapid in order to allow for many such analyses to be carried out in as short a time span as possible.

The result of this trend has been that the chief duty of the computer has been as a text handling system which must selectively print out, for the student at his terminal, various messages depending on how he answers questions from time to time.

I choose to call such a system a "page turn" 'ing system. One of the most important characteristics of the "page turners" is that by their very definition they cannot encompass very much flexibility. In lessons of this type one measure of the sophistication, and of the worth, of the lesson is the number of different student responses that the instructors anticipate and provide for. Most systems provide for automatic collection of unanticipated responses in order to allow the authors to improve the performance of the lessons at a later date.

However in any such lesson it is clear that only a finite, and very small, number of possible answers can be anticipated. Furthermore, even if more answers can be anticipated, after a very short time the point of diminishing returns is reached, beyond which it is no longer worthwhile, in some sense, to provide for more possibilities. This in itself is an objection to "page turning" systems. It can be seen that it makes them relatively inflexible.

There is an even more important objection. In most subjects the most important thing that a student learns is how to solve a particular type of problem, whether the problem is translating from one language to another, or finishing a mathematical proof. In learning to solve problems the student needs a great deal of individual assistance from someone relatively skilled in the discipline that he is learning. It is at this point that most instructors hope that the computer, and CAI , will be of some assistance to them. The computer, to be most useful, should be able to fill the role of a tutor who helps the student solve problems in the field in which he (the student) is interested.

"Page turners" have two difficulties in filling this role. First they cannot help students with their questions. Rather they can only do specific examples for which the teacher has written a lesson. This very specificness is, in itself, a limitation. Every student will have his own question which is particularly difficult for him, and perhaps easy for most of the rest of the class. Unfortunately with "page turners" it is not possible for the student to ask for a particular example unless it happens to be one that the teacher has written a lesson for.

There is another, possibly more serious objection. In a page turner it is possible for a student, if the lesson is well written to alter some of the details of the solution. However it is not possible for him to alter the actual method of attacking the problem. If the student is continuously exposed to one persons method of solving problems a sort of insidious brain washing can occur. This is actually a peculiarly human characteristic because the human is essentially a problem solving animal. Thus if one can train him to solve problems in a manner sufficiently similar to some other person he becomes, himself, similar, in the way he thinks, to that other person. From a pedagogical point of view another problem occurs. Not every person is capable of solving all problems from every point of view with equal facility. Thus the particular method of attack on a problem proposed in a given lesson may not be suitable for every student using the lessons, and there is even a chance that it will not be suitable to the majority of the students.

It is hoped that the package described herein solves some of the problems explained above and thus is a general advance in the field of CAI.

It may be noted that many proofs, in mathematics, are not dependent on a particular point of view. They may be considered to be independent of the basis from which one views them. This is, of course, not true of all proofs. One of the characteristics of mathematics that makes it so "basis independent" (in the sense described above) is that it is deliberately constructed so that the system, considered as a complete system, has as few assumptions as possible. This is not true for every discipline, or even for all branches of mathematics.

If one could express the substance of a discipline in mathematical terms, or in more general, but still "basis independent" terms then the material would be less "basis dependent". What had been expressed in this way could be explained in some way which would not have to depend on the student's viewpoint.

Many subjects, including most of the sciences, are very heavily dependent on mathematics. Even more important much of the material involved in these subjects can be expressed in mathematical terms. By doing this one can make these subjects much more ""basis independent"". In the case of philosphy, if one expresses all of the arguments in the symbolism of propositional calculus then the arguments, qua arguments can be dealt with separately from the statements of which they are composed. In this way one can dispose

FILMED FROM BEST AVAILABLE COPY

of part of the problem of attempting to prove the correctness of a particular argument by reducing the problem of deciding whether or not certain statements are true.

Unfortunately this reduction is not always a simplification but in many cases it is. Thus for students it can help them see the truth or falsity of a particular argument and the conclusions drawn from it.

In any case the last few paragraphs were just an interesting aside. The point brought out in them is that the computer can be used in one half of the process described above. The process involved deciding if a particular argument and the conclusions involved were true. We can, then, divide the decision making process into two parts which can be made quite distinct. The first step is to decide if the argument, qua argument, is correct. This can be done, in many cases, with the help of the computer. Then it is necessary to decide if the statements of which the argument is composed are true. This can not always be done with the help of the computer, and in most cases probably cannot be done at all by or with the computer. The second step is also complicated by the fact that it is not always soluble.

This paper describes one system for helping students decide if a particular argument is true or not. It is important to remember that the argument is considered, as an argument, abstracted from the truth of any of its constituent statements. Thus the programs developed, considered as lessons, are relatively "basis independent". Thus if there is any "brain washing" going on it is, essentially, being done directly by the teacher, and not by the computer. More important the lessons are applicable to all students and not just those with a particular point of view.

The discipline for which the package is intended is symbolic logic. I do not call it by the more general name Propositional Calculus, or Predicate Calculus because the system considered does not contain any quantifiers.

The purpose of the system is to help the student learn the "mechanics" of the proof procedure involved in symbolic logic.

For many years symbolic logic has been difficult to teach succesfully. The problem, basically, is that the students, particularly those without any mathematical sophistication, are not familiar with the type of thought required in order to be able succesfully to understand the elements of logic. What they need more than anything else is opportunities to engage in plentiful practice with the help of someone familiar with the subject material.

Ideally this would mean that a tutor would sit down with each student and help him in doing proofs. The tutor would critisize the students proof s as he was doing them and would continue to help in this way until he was satisfied that the student understood the material well enough to continue on his own.

Actually the situation outlined above is the ideal for nearly all subjects and disciplines. Unfortunately it is not realised in very many, if any, subjects. The most important difference between symbolic logic and other subjects is that the students either can't or won't learn the subject in any other way. This is supported by the fact that most students consider logic to be an extremely difficult subject and that few of them really are able to do ever the formal manipulations involved in the proof procedure succesfully, let alone, really understand the basic ideas behind these manipulations.

If I might hazard a guess as to why this is the case it might be due to the fact that most beginning university sutdents are not used to the self discipline and constant study and practice involved in learning mathematically related subjects. Even "pure mathematics" is a problem oriented discipline in which the novice learns by doing problems. Thus because they do not have, or realise that they need, the self discpline required to learn logic they do not master the subject successfully.

However one should not be too quick to apportion all the blame to the students. The learning of the ability to do problems is not a subject amenable to lecture treatment and this is a large contributory factor to the low success rate in true understanding of the concepts of logic.

The package described, then, "simulates", after a fashion, a tutor for symbolic logic. It is meant to fulfill the needs described above.

We expect that it will be used, in its first complete version, this year.

The remainder of the paper is devoted to describing the package itself. Few technical details about the programs are given since these would needlessly obscure the concepts and principles of operation explained. Further details are available from the author on request.

BASIC OPERATION

This section is devoted to explaining the package as the student sees it when he sits down at a computer terminal.

The terminal itself is an IBM 2741 typewriter terminal. Physically it is similar to an IBM selectric office typewriter. In principle, however, any terminal which is supported by the APL system can be used. While, at present, this represents a fairly limited set of terminals the situation is rapidly improving.

The student can type in answers and questions, as described below, and the computer can read what he types. The computer can, also, type out whatever it likes at the typewriter for the student to read. Thus a "conversation" can be carried out.

Let's assume, for the moment, that the student has a question to ask. Since the package is restricted to symbolic logic the question must be expressed in the "language" of symbolic logic. This means that the question must be in the form of a theorem to be proven.

The student sits at the terminal and identifies himself to the computer. This is just a formality which he must go through in order to use the computer and to let the teachers know who is using the programs.

He can then type in the theorem. At this point there are two possiblities. The student may already have a proof which he wants the computer to "look" at; or the student may not be able to do the proof and wants some help from the computer.

In either case he begins to type in as much of the proof as he knows. As he types each line the computer looks at it. If the line is correct the computer allows the student to continue. If the line is incorrect the computer attempts to help the student.

In order to understand better how it helps the student let us consider for a moment what a line in the proof may look like. Each line consists of a formula, expressed in the symbolism of symbolic logic, and of a justification which is supposed to explain why the line is there.

If the line is correct then this means that only the justification is incorrect. This may be due to one of two reasons. The justifications may be incorrect due to the fact that the student has left out several lines in the proof. Or the justification may be totally in error.

Similarly the formula may be correct, and the justification also correct, but the student may be heading in the wrong direction.

Finally the formula may be totally "wrong": it may not be a valid formula, or it may not be derivable from the preceding lines.

In each of the above cases the computer attempts, by the device of asking the student questions, to show him where he is wrong. In the case that the student is going in the wrong direction the computer allows him to continue until he works his way into a "dead end" and then points out to him by asking him questions about each of the various directions in which he may continue why he cannot reach the goal. Then it shows him where to go back to.

If his line is totally wrong it reviews the relevant rules of the logic with him until he finds where his mistake is.

It proceeds in a similar manner in each of the remaining cases.

It may happen that in the process of asking the student questions the computer decides that there is a fundamental misunderstanding of a concept. In such a case the computer will send the student to see one of the instructors of the course.

Until this point we have assumed that the student has a question, in the form of a theorem, to ask about. Often, however, the student does not know what questions to ask. In such a case the computer will ask him questions and help him solve them.

The computer must have some indication of his ability, and this can be gained by asking him what rules of inference of the logical system he is familiar with.

Then as the student answers the questions posed, by himself or the computer, the computer can observe which rules of inference he has trouble dealing with and concentrate more heavily on questions dealing with those particular rules of inference, than on the others, when asked to pose questions. Thus the computer can concentrate on the students weak points.

At this point it will be seen that the system, as a question answering system, is complete. It can answer questions dealing with any theorem in symbolic logic with certain rather arbitrary, and adjustable, restrictions on the length of the formulae dealt with.

In addition the package can help the student begin by posing questions which will moreover be aimed at his own difficulties.

Each instructor must now complement the package with more conventional lessons of his own. These deal with the more formal definitions which the student must learn before he can begin doing proofs. A package of lessons is provided, more for the sake of completeness, than for any other reason, and is self sufficient. If any instructor does not like the treatment he can, with very little effort, write his own set of lessons to go along with the question answering system.

Note that the type of question that the student may ask is quite general. If he is, for example, having difficulty with one particular step in a proof he can express this step as a theorem to be proven and thus ask the computer to help him.

The actual notation used in the symbolic logic can be specified. Two forms are possible: Polish; and Infix. In polish notation, also know as "Prefix" notation, operators always occur to the left of their operands. In infix notation operators always occur between their operands. The cheif advantage of Polish notation is that brackets are not necessary and for this reason it is more suitable for work with the computer. Infix notation, on the other hand, is easier to read and is more conventional although it does require brackets.

In addition to being able to specify the type of notation the instructor can also specify the characters to be used to represent the logical operators. Since the typeball on the terminal can easily be changed virtually any characters can be used. The typeball usually used for APL is particularly suitable because it contains all the characters used in most textbooks to represent the operators. It, in fact, has all the characters used for most operations involved in set theory and thus is a very satisfactory choice.

In addition to specifying the characters to be used to represent the operators the instructors may also specify which operators will be available to the student in his proof by specifying what rules of inference may be used. Thus the instructor may specify particular logical systems (i.e. intuitionistic/classical). The student is then only allowed to use the rules specified in his proofs.

Finally the instructor may specify the names used for each of the rules of inference. This last is just a question of taste and will allow the proofs to conform to any particular text used with the course.

THE PROGRAMS

All of the programs are written in APL (A Programming Language). As such they may be used on any computer which supports APL. The only changes from system to system would be minor because of the uniformity of APL as a language.

The complete package fits into one APL workspace (32K). The accompanying lessons and the response analysis for the questions that the programs ask the student when he has difficulties are handled by programs from a package, written in APL by the author, called APL/CAT (APL Computer Assisted Teaching package) which has been in wide use for about one and a half years.

The most important program in the package is a "Theorem Prover". This program can, given a theorem, produce a proof in a very easily readable form.

Each proof consists of separate lines. On each line is put one formula and its justification. The method of proof is that of subordinate proof. Paragraphing is used to make the subordinate proofs easily distinguishable: each subordinate proof is indented two spaces from the "encompassing" proof. All the justifications form a column on the right of the proof. Each justification consists of one to three items. Always included is a rule by which the item may be justified. Then zero, one, or two line numbers of preceding items which may be involved in the justification are given as may be required.

Each line is numbered in sequence and a margin is left between the number and the item. In general the theorem being proved is not stated at the top since it is always the last line of the proof.

Actually the student never comes into direct contact with the theorem prover. The program that he comes into most contact with is the "proof checker". As he types in a proof a line at a time it checks his proof to see if it is correct. First it sees if each line is a "WFF" (Well Formed Formula). Then it checks to see if the line does indeed follow from the given justification.

If it does follow from the justification and is a WFF then the student is allow to continue. If the line is not a WFF the program helps the student analyse the expression he has typed in. Essentially it analyses the expression with him character by character until he sees where his mistake is.

If his line does not follow from the justification the theorem prover is called. It is given the proof to date, which must be correct, since it has been previously checked, and asked to attempt to derive the student's line. If it can do so then the justification is in error but the formula is correct. If extra lines are required to derive the formula then the student has skipped over several steps. If the formula cannot be derived then the student has typed in an erroneous formula.

If the student cannot continue, because he is unsure how to proceed, then the program asks the theorem prover what the next line should be and by way of questions helps the student find it.

Finally when the student finishes the proof the program calls the theorem prover to determine if the proof is as concise as it might be. If not the program helps the student see why it is not. Generally if the proof is not as concise as it could be it is because the student went in the wrong direction for a while. Then he either realised what had happened, or got stuck and got help from the computer. Thus the computer ensures that the student is aware that he went in the wrong direction for a while.

There is a series of programs that serve as interfaces between the two programs described above. These convert between the various notations used internally, and those specified by the instructor. The notation used by the Theorem Prover is a modified prefix (polish) notation.

Another important component of the package is the "Question Asker". This program will, on demand, ask the student a question which he can try. When selecting the question the program uses two criteria. First it asks a question which only needs the rules of inference, that the student has said he knows, to do. Secondly, every time that the student makes an error, in the checker, this is noted by the computer along with a note as to which rule of inference is involved. The Asker then asks more questions about this rule of inference. The "more" used, previously, is a statistical "more". The questions are "randomly" constructed in such a manner that the rules of inference that the student has difficulty with are asked about more often than others.

These, along with a few support programs, comprise the entire package. As was said the technical details will not be discussed.

POINTS OF INTEREST

A: WFF'N'PROOF

The package was constructed in such a way that, if the instructors specify polish notation, and specify the characters used in the games WFF'N'PROOF, and the names of the rules of inference used there, then the package can be used, in a very natural way, to help teach the games. The students can also then use the computer as a sort of referee for games since it can prove theorems when required to do so.

It is almost certain, in addition, that the computer could play the games with the student, and, indeed, programs are being written to do this.

B: OUTPUT OF PROOFS

The package has been deliberately constructed in such a manner that the proofs generated by the theorem prover are never output to the student. This would correspond to "giving" the student the answer. Instead the other programs in the package use the proofs generated to ask the student questions to help him find his difficulty.

It is quite possible, however, to have proofs output by the program. Thus the theorem prover can be used by itself, if desired, to produce proofs of theorems.

C: DETERMINATION OF PERFORMANCE OF STUDENTS

As was mentioned earlier the package uses the APL/CAT package. This means that the teacher may select to monitor the student's performance to any degree that he likes. He can also monitor, by the same token, the performance of the package.

D: INDEPENDENCE OF TEXT

In addition to the options mentioned above which the teacher may change he can also change the questions which the computer asks the student when he has difficulty. Thus the teacher may make the package as indenpendent of any particular approach as he pleases. Since the package fits comfortably into a workspace there may be several versions of it available at any given moment and the students may select the one most appropriate to their particular course.

THE DESIGN AND EVALUATION OF AN ADAPTIVE TEACHING SYSTEM

J. R. HARTLEY

Research Councils' Computer Based Learning Project
The University of Leeds
ENGLAND

THE DESIGN AND EVALUATION OF AN ADAPTIVE TEACHING SYSTEM

SESSION 7
ROOM/SALLE 148

By/Par: J.R. Hartley,
The University,
Leeds, England.

The term "adaptive" is used to imply that the computer generates tasks to fit a pupil's competence, as well as supplying individual feedback and remedial exercises based on patterns of error. The system itself is given a representation or model of the task, builds up a student image from his performances, and has a series of teaching operations it can employ under the control of means-ends guidance rules. The teaching potential of such systems is much greater in certain contexts than those which use pre-stored material, but more precise information about task and student has to be uncovered through systematic experiment. These approaches have been developed at Leeds in Junior Mathematics and Medical Diagnosis.

A series of experiments will be outlined. These investigate models of difficulty in some arithmetical and diagnostic tasks, and study how adjusting task difficulty to fit student competence, and varying type of feedback and remedial exercises, affect learning gradients. Also, in the medical area a computer based diagnostic game was devised to examine a student's strategies of choosing and processing information. These experiments suggested models and guidance rules to be used by the computer in its adaptive teaching system.

It is argued these systems not only have greater teaching potential but provide a general framework for systematic experiment and theoretical discussion.

LA CONCEPTION ET L'ÉVALUATION D'UN SYSTÈME PÉDAGOGIQUE ADAPTIF

Le terme "adaptif" signifie que l'ordinateur propose un travail du ressort de l'élève tout en fabriquant des "boucles" pour les réactions et des exercices de correction basés sur des modèles d'erreurs. Le système lui-même reçoit un modèle du travail et des exercices, reflète l'élève selon sa compétence et dispose d'opérations enseignantes qu'il peut employer selon des règles donnant les fins en fonction des moyens. Le potentiel pédagogique de ces systèmes est beaucoup plus grand dans certains domaines que le potentiel des machines se servant de données mises d'avance en mémoire, mais des informations plus précises sur le travail à effectuer et sur l'élève doivent être trouvées à la suite d'expériences systématiques. C'est là le résultat de notre travail à Leeds en mathématiques pour les jeunes et en diagnostic médical.

Nous allons décrire une série d'expériences permettant d'étudier des modèles de difficultés recontrées dans le diagnostic et en arithmétique, aussi d'adapter ces difficultés au niveau des élèves et de faire varier le type de "boucle" et des exercices de correction affectant la vitesse à laquelle l'élève apprend. Nous avons aussi, dans le domaine médical, mis au point un jeu dit de diagnostic visant à déterminer pourquoi et comment un élève choisit et traite l'information. Ces expériences nous ont amenés à définir des règles directrices et des modèles que l'ordinateur doit utiliser dans son enseignement adapté.

Certains avancent l'idée que ces systèmes, non seulement ont un plus grand potentiel d'enseignement, mais aussi fournissent un cadre général d'expérimentation systématique et de discussion sur le plan théorique.

1. INTRODUCTION: THE GOALS OF COMPUTER ASSISTED INSTRUCTION

The innovations of Educational Technology have been disappointing in the sense that they have had only limited effects upon educational practice. However, this does not appear to have blunted the enthusiasm of some workers in the field who tend to greet each new and expensive advance with the same over-optimistic hopes of achievement and benefit. (See illustrations by Rubinoff 1971.) The reasons why these promises are not fulfilled do not lie within the nature of the innovations themselves. Educational Technology includes the production of material which is to be displayed and managed by devices, and it is clear that many other of its characteristics, for example, behavioural specifications of objectives, task analyses, and close monitoring and evaluation of responses, correspond to sound pedagogy. Oettinger (1970) gives his views as to the reason when he describes some characteristics of the practical teaching situation and the administrative structure of education, its inertia and complexity, and the difficulties which must arise when any new major technique has to be incorporated within it. No innovation in educational technology has made an explicit and determined attempt to solve this problem, although some projects which use the computer (e.g. Pittsburgh: Ticcit scheme, Texas) seem to be attempting partial solutions. However, in respect of CAI Oettinger's viewpoint may not account for all the difficulties.

The aim of 'institutionalising' the computer so that it plays an integral part in administration, teaching and research activities in ways which are cost-effective must be a long term aim. Moreover, many problems must be solved before this aim is realised. This realisation involves the development of large scale multi-access systems, computer and author languages, studies in curriculum development and much more precise experimentation in ways of controlling learning. It could be argued that the present state of knowledge in Psychology and Computational Science make any aims of useful application of CAI unrealistic, and that resources would be better employed in using the computer as a laboratory for studying learning and for developing methods of representing student performances and types of tasks. This argument (e.g. Clowes 1971) maintains that author languages are primitive devices which necessitate all the educational reasoning and methods of decision making being carried out by experts prior to the teaching material being written. Until techniques have been developed by which the computer can take over this responsibility and manipulate knowledge structures using natural language communication which goes beyond keyword matching, the objectives and the learning gains made by students are likely to be limited, and will not justify the effort which has to be put into the production of educational material.

These arguments have some force, and work in Artificial Intelligence, which is tackling these problems, could have beneficial and important effects on CAI. However, it is only by setting projects in practical teaching areas that the nature of the difficulties of managing learning will become evident. It is also clear that present facilities in communication and data collection do allow proper experiments to be carried out. Therefore, although the CAI project at Leeds was set up in 1969 by a three-year grant of £120,000 from the Science Research Council and the Social Science Research Council, and although the project was oriented to research rather than to provide service teaching, the long-term underlying aim has been a practical one, namely institutionalising the computer as a teaching device within a school and within some university departments.

The research moneys provided for staff and for the purchase of a small computing system entirely dedicated to experiments in CAI. The system includes a Modular One computer with 16K fast store, a 1M word disc and some 12 terminal stations which are teletypes and audio-visual devices, some of which are remotely connected through Post Office lines. The main emphasis of the research has been to develop adaptive teaching methods where 'adaptive' is not used in the sense of merely providing different feedback comments or allocating students to different remedial material. It is used to indicate that the student is routed to different teaching approaches or that different tasks are put together as required on-line to fit a particular student's competence.

There were several reasons for taking this approach. First, although limitations of the terminals restrict the communication between the student and the computer, the record keeping facilities and fast operating speeds which are available enable it to be a powerful decision maker. Provided that the representations of student performances and tasks can be made in precise and quantitative terms, the computer itself can be used to develop a theory of instruction as a prescription of rules which specify the most effective ways of acquiring knowledge or skills. This experience can then be applied in on-line decision making, and should have a significant pay-off in terms of learning gains. Second, if the computer is to be used to supplement the work of the teacher in those applications where it is difficult for him to give individual attention, adaptive methods, of the type defined above, are likely to be important. In this respect it seems useful to distinguish between teaching functions of exposition, in which objectives stress acquisition of knowledge and its retention, practice, in which skills are repeated in context up to prescribed standards of mastery, and what might be termed '<ins>problem solving</ins>' in the sense that the material has some novelty and the student is expected to apply acquired knowledge and skills in a solution of his own devising. In exposition, where objectives tend to be standardised over a whole class, or over smaller groups of students, other methods seem more appropriate. Our view has been that greater individualisation of control is needed in the latter two areas and these are the ones in which the computer can be employed with great benefit. Finally, it seems sensible to tackle particular teaching areas in which conventional methods are meeting with problems. For these reasons, work at Leeds has concentrated on the specific subject areas of: Junior School Mathematics (age 7-11), the teaching of English to immigrant children, University Chemistry, and Applied Statistics in the Social Sciences. These areas provided the computer with a range of different types of teaching problems, and also gave opportunity for providing useful supporting facilities of calculation and simulation.

2. THE FRAMEWORK FOR ADAPTIVE TEACHING

The aims of an adaptive teaching system stress the optimisation of instruction by relating teaching methods to the type of learning task and profiles of students' performances. Almost ten years ago Smallwood (1962) showed how the computer, by storing student routes through a program, could use comparison methods to learn to guide new learners individually through the teaching. Although no evaluation of the gains of this adaptive programming was made, if the task is such that "models" of learning can be inferred from student performances (i.e. a quantitative estimate of the state of the student can be made from performance, and the effect of teaching method on that state is known) optimal decisions can be made (Laubsch 1969: Van der Veer 1970). These schemes suggest a "problem solving" approach to the management of learning. Initially the student is in a learning state represented by pre-test information, and this has to be improved and altered to the state represented by program objectives. Various teaching operations can be carried out which can analyse the task and set subgoals, sequence material in different ways, and provide help and feedback on the basis of responses. Previous experience and performance profiles allow means-end guidance in the selection of these operations to be given continually. The components of this view of instruction are:

(i) A representation of the task.

(ii) A representation of the student and his performance.

(iii) A vocabulary of (teaching) operations.

(iv) A pay-off matrix or set of means-end guidance rules.

The representation of the teaching task must have two components. A task analysis can indicate the ways in which content is arranged but in some areas, such as arithmetical practice tasks or medical diagnosis which will be described later, rules can be given which allow the computer programs to generate material by choosing elements and composing them into a task. The second component is another set of rules by which the teaching programs can recognise different classes of task, for example by having measures which describe difficulty or complexity. This will enable equivalent tasks to be made up for practice, similar forms of the same task for giving wider experience, or more difficult or easier tasks to match against a pupil's competence. Much of the representation of the student will be built up from on-line performance data, both to give general levels of competence and to allow specific hypotheses of types of error or misconceptions to be tested. To allow the number of teaching decisions to be reduced to manageable proportions, the teaching operations should be hierarchically arranged. They include (i) the sequencing of tasks to provide a curriculum and within this (ii) the teaching mode, e.g. the broad method of control, whether directive or inquiry, within this (iii) the criterion level before help or prompting is given, or remedial exercises stipulated and (iv) the type of feedback, prompting and help which are allowed.

While the representation problems are difficult, there is a severe lack of knowledge about the means-ends guidance rules. What is needed are experiments which will show the effects of the teaching operations on the learner's state. In some cases the models of learning will indicate which teaching presentation is necessarily optimal over all students (Atkinson and Paulson 1970) but in general (and in the arithmetic and medical projects to be discussed later) these conclusions will differ over groups of students and the amount of time which is available for teaching.

The advantage of this formulation of adaptive teaching is that it gives a framework for systematic experiment and for more precise descriptions of performance and learning. This will be illustrated by referring to two specific projects, the management of practice tasks in Junior Arithmetic and at the university level, problem solving tasks in Applied Statistics and Practical Chemistry.

3. PRACTICE TASKS IN JUNIOR MATHEMATICS

The experiments which will be discussed concern arithmetical calculations set in a vertical format. However, this material can be used in conjunction with other exercises which need application of, and give insight into, the properties of the numeration system. These take the form of games played at the computer terminal which have the children decompose and recompose numbers in ways which illustrate the ideas of place value, equality and associativity.

Practice in computational skills has tended to be overshadowed by recent advances in other areas of elementary mathematics teaching. Nevertheless skill in computation is essential and such practice can be managed both economically and efficiently, and quickly shows significant improvement in performance (Woods et al. 1969). The Leeds programs have three features of interest. First, the computer keeps individual records which are accumulated during (past) sessions, together with a marker which indicates a pupil's present position in a lesson. A series of lessons (a lesson is a set of similar tasks) can be set against a student's name and he works through them until specified criteria are met. Second, the computer generates lesson and remedial material. A lesson specification has parameters or constraints which determine the form and difficulty of the questions. These parameters may be specified directly by the teacher, may be modified by using a library of stereotyped examples, or may be constructed by the computer using 'models' collected from previous performances. The numbers used in the questions are generated randomly by the computer and filtered to these constraints. The simpler remedial questions are also generated but use numbers which are based on particular errors the student has made. The third feature of the Junior Mathematics Program concerns the methods of conducting the teaching with pupils. In general, the question is printed on the teletype and the pupil types the digits he considers to be correct. Incorrect digits are not printed, instead a bell rings and he tries again. Further errors produce feedback (of various types) which is given to assist him to reach the correct answer. The criteria for moving between and mixing question types, and for entering remedial sequences can be specified by the teacher, for each pupil, or the computer can decide this by consulting its means-ends guidance rules. Thus the teaching operations adapt in three ways. (i) By generating tasks which match a pupil's competence (ii) by providing different types of feedback and (iii) by generating remedial material.

3.1 Models of Task Difficulty

A first set of experiments was to represent the task using hypothesized models of difficulty and to relate these to the actual levels of success which pupils showed in their working. The method is general for all tasks which can be represented as a sequence of operations. An error factor is associated with each of these and the probability of success can be equated to a polynomial in which the error factors appear as constants for the individual or group of students considered. For example, in considering the addition of a column of numbers (and ignoring in this illustration 'carrying' conditions) the model assumed the pupil undertook a mental ordering corresponding to the natural number sequence. Addition of one to a number thus became a matter of moving a pointer (indicating the partial sum at that time) once along the sequence. Adding a general number to a partial sum therefore has three components:

(i) The pointer is pre-set to a value equal to the number (the partial sum) held in memory

(ii) The operation of moving the pointer is repeated a number of times which corresponds to the number to be added to the partial sum

(iii) The answer (the new partial sum) is stored in memory

This loop is repeated until there are no numbers left to add. Such a scheme is illustrated in Diagram 1

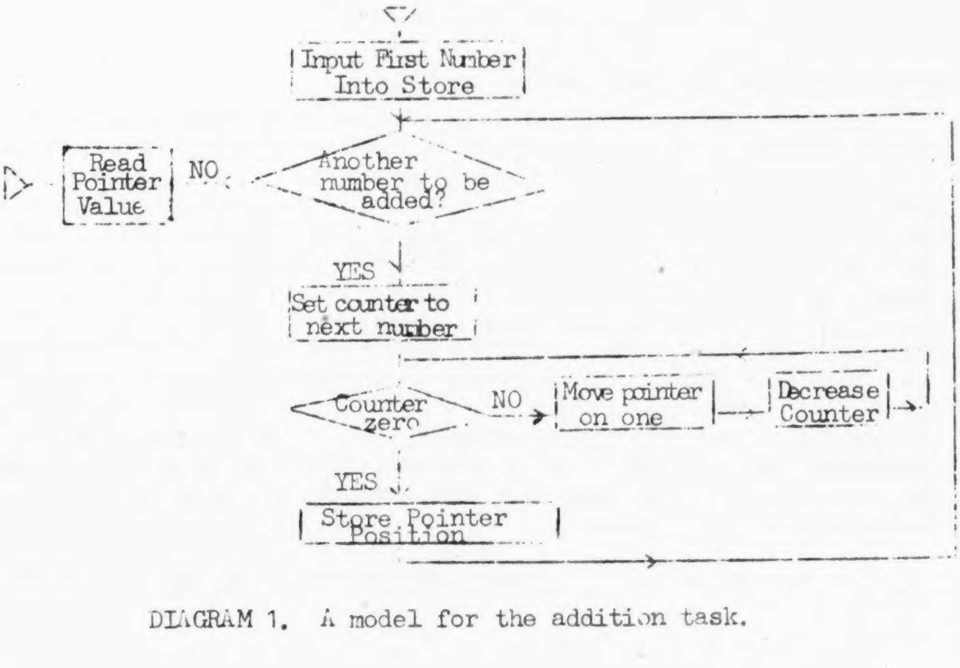

Input First Number Into Store
Another number to be added?
NO
Read Pointer Value
YES
Set counter to next number
Counter zero
NO
Move pointer on one
Decrease Counter
YES
Store Pointer Position

DIAGRAM 1. A model for the addition task.

The work of Suppes (Suppes 1967) gives some justification for this approach, but if the model is developed and E_1 and E_2 represent error factors associated with 'pointer moving' and 'storing and reading partial sums' respectively, a formula for the total probability of success of adding a column is obtained. If certain assumptions of independence between operations is made this reduces to: $\log p(s)/_{r'} = s' \log (1-E_1) + \log (1-E_2)$ where r' is the number of rows and s' the average number size. To validate the model, pairs of exercises, which had the same numbers set in exercises of 1 and 3 columns were generated at 15 levels of difficulty. Approximately 200 pupils worked through 240 examples each (properly randomised to control presentation and sequence effects) and comparisons were made between student ages, difficulty level and types of errors, using analyses of variance and least squares curve fitting techniques. The data was extremely well ordered, gave support for the difficulty model and allowed estimates of error factors to be made for various student groups. These methods were also extended to other numerical operations e.g. subtraction where several plausible models can be stipulated depending upon the strategy followed by a pupil. The methods of incorporating these models for use in generation of tasks which increase in "difficulty" as a learner becomes more competent have been described elsewhere (Woods & Hartley 1971). They use as a controlling parameter the probability of success at which it is stipulated the individual should work. The pupil representation thus includes his actual working level of success and his error factor estimates (E_1, E_2, etc.) which are associated with the operations he [illegible] using in the calculations.

3.2 Some Experiments in Adaptive Teaching

In order to study the effects of working at different rates of success an experiment was carried out in which pupils were randomly allocated to one of three treatment groups after being given a pretest. These groups had the difficulty of the questions altered so they worked at levels of success 95% (high) 75% (moderate) and 60% (low). Post-tests were given at the conclusion of the treatment. Overall the results showed a curvilinear relationship with the 'moderate' level group having the advantage in terms of speed and accuracy over the 'high' level which in turn had a distinct advantage over the 'low' group. A three-way analysis of variance using pretest level, treatment groups, and difficulty of item in the post-test as factors showed finer differences. Significant interactions indicated that treatment level had most effect on those low on pretest scores, i.e. the less able were more sensitive to differences in treatment.

The second method of adaptation concerned the type of feedback which could be given to pupils. Knowledge of results is usually held to be important in learning, although many experiments have not succeeded in demonstrating its beneficial effects. An experiment was undertaken in which all groups worked at the computer. The difficulty levels of the questions were controlled, but the treatments varied on feedback conditions. The exercises involved multiplication of a 3 digit number by a single digit. In one group no information was given other than at the end of the session the pupils were told how many questions they had correct. A second group had 'passive' feedback, so that if a second attempt at a response was incorrect the correct working was printed out. A third group worked with 'active' feedback in which a sequence of questions and answers broke down and explained the steps involved. The pupil then repeated the question again. This last condition enabled the sources of pupil errors to be diagnosed and classed as 'memory overload', process, transposition, and conceptual errors.

Initially the pupils were given a pre-test, randomly allocated to treatment groups and after six working sessions of fifteen minutes given a post-test. The results were moderately encouraging. Briefly, the trends were in the predicted direction and the overall effect was significant. . The no feedback group had no improvement at all, the 'passive' feedback group were better, and the 'active' feedback group scored highest. The experiment seemed to suggest that for low performers 'active' feedback is best, and the more detailed error analysis which this gives can indicate the appropriate type of remedial exercise. As a person becomes more proficient passive feedback is probably sufficient and this allows more questions to be completed in the same time. More precise experiments relating to the effectiveness of feedback are now in progress.

The computer programs allow hypotheses about types of error to be tested on-line. Associated with each hypothesis is a test routine which processes each response a student makes. The results become part of the student representation. Also the numbers with which errors have been made are stored. This information indicates the need for particular types of remedial exercises. A simple experiment with multiplication tasks set in vertical format was carried out to see if persistent process errors could be eradicated by generating remedial questions based on his errors. (A persistent error is one in which an incorrect response is given to the same question on several occasions.) The difficulty level of questions was again controlled and there were two treatment groups - with and without the remedial facility. Depending on the frequency of a pupil's errors the remedial exercises were interspersed in the main practice sequence. Examination of on-line error analyses showed a successful result with the remedial group having a much lower proportion of persistent errors after six fifteen-minute working sessions, but the actual number of persistent errors was few.

The results of these experiments give some useful guidance rules and these are being integrated into a fully adaptive teaching system. This cannot be considered as a self improving system (beyond refining the accuracy of its estimates as more data is accumulated) since it does not alter its teaching strategy for the individual if performance expectations are not fulfilled. This is for future development. During the last six months the package has been used by teachers, and integrated within the classroom. This is important because these programs merely guide practice using operations and concepts which have been taught previously. Moreover studies (Abramson et al. 1970) show much benefit is lost if close reference is not made between computer and teacher. The work at Leeds first used the package diagnostically, to provide the teacher with performance standards of individual children; then it was used in parallel with new teaching, and finally it was used for revision. The teacher set down the performance standards he would normally expect at the end of one year, and in fact these were accomplished in a little over three months. Therefore, although this control of practice might be educationally limited, it can be a significant help to the teacher.

4. PROBLEM SOLVING TASKS IN APPLIED STATISTICS AND PHYSICAL CHEMISTRY

The previous illustrations in mathematics although allowing the computer programs to take a greater part in decision making and to generate material when it is required, are educationally small. However, for many areas of the curriculum which are more prominent in teaching but which are less well structured, it is much more difficult to develop adaptive methods. At Leeds we have become interested in two such areas at the university level, namely teaching students to plan and execute practical experiments in Physical Chemistry and Statistical problems in the Social Sciences. Both these tests are similar in that they involve relational types of problems in which a dependent variable is a function of a set of independent variables. In some exercises the students have to establish the effects of the independent variables, and in others establish the particular function which relates them to the dependent variable.

The practical difficulties of teaching Applied Statistics are well known and much discussed. The classes tend to be large, there are wide differences in the students' previous mathematical experience and many of them feel unenthusiastic when asked to work in quantitative and abstract terms. Practical Chemistry has its difficulties too. Much time is spent in the laboratory, but the rota method of organisation means that the theoretical work of the lecture room is difficult to synchronise with the practicals. The instructions given to students tend to be detailed and directive and do not specifically teach the student to plan his work.

Over the last twelve months we have used the computer to help second year Chemistry undergraduates to plan their experiments. After formulating their plans, they collect data by using chemical apparatus, but in Statistics where similar methods are followed, the collection of data is done by using simulation techniques. An elementary analysis of these planning tasks makes distinctions between (i) a planning algorithm or procedure which has four parts: setting the goal, the design of methods of data collection, the treatment of data, and contingency planning, (ii) the (student's) knowledge of the subject and the operations which are involved in executing the plan, and (iii) the application of the planning algorithm to the (student's) knowledge base to solve the particular problem in hand. The teaching material is pre-stored and is written using an author language. It follows the distinctions outlined above and can be adaptive in the choice of overall teaching strategy, the amount of help, and the particular remedial sequences which are given.

4.1 The Teaching Methods

The teaching operations have two general features of interest. First, there is the development of teaching strategies which put different amounts and types of control on the student. In the 'strand' method the computer always leads, and, depending on progress, moves the student periodically between small, medium and large step 'strands'. In the 'file' method, the teaching control follows a mixed initiative pattern. Information (on planning, facts, apparatus details and definitions) is stored in files which are structured and from which the student can collect the information he feels he needs. Within certain limits there is free movement between files, but, finally the student must enter a check file to ensure his plan meets certain specifications. In the check file the computer assumes complete control, no external files can be consulted, and if progress is not satisfactory the student is routed to the appropriate sections of planning or given remedial instruction. The structure of the program is shown in Diagram 2. The supposed advantage of the file system of teaching is that it requires the student himself to decide which information or type of help he needs and so he must continually evaluate his progress and analyse his difficulties. The adaptive teaching system should move a student between these teaching modes,

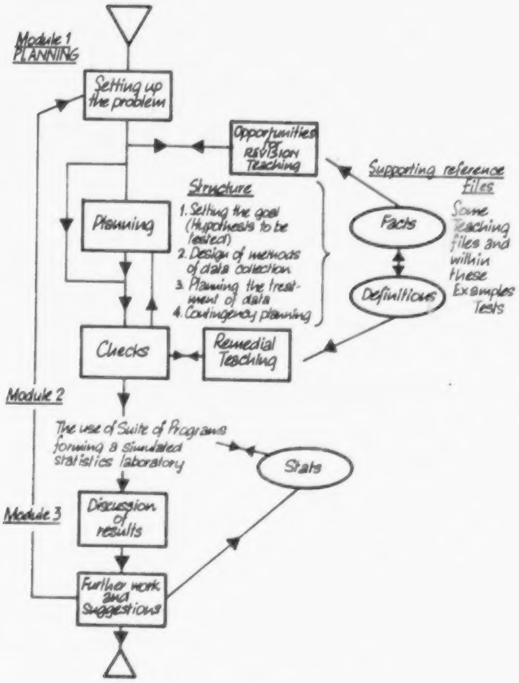

Module 1
PLANNING
Setting up the problem
Opportunities for REVISION Teaching
Supporting reference Files
Structure
1. Setting the goal (Hypothesis to be tested)
2. Design of methods of data collection
3. Planning the treatment of data
4. Contingency planning
Planning
Facts
Definitions
Some Teaching files and within these Examples Tests
Checks
Remedial Teaching
Module 2
The use of Suite of Programs forming a simulated statistics laboratory
Stats
Module 3
Discussion of results
Further work and suggestions

DIAGRAM 2
The structure of the computer based teaching programs.

selectively fade the help he can obtain from the files, and also present him with a proper sequence of problem solving tasks. The information for this decision making depends heavily on the representation of the student through his performances, and through hypotheses which the computer builds up about his knowledge of the subject area, the planning algorithm and his competence in its application

The second feature of the teaching is the use of simulation techniques, particularly in Statistics, where a simulated statistical laboratory has been developed. This suite of computer programs gives the student facilities for setting up population parameters, for having samples drawn from them which he can edit, and for having specified calculations carried out on data. For example, the student might wish to enter sets of population parameters at the teletype. The computer asks if the required distribution is standard (e.g. Normal, Binomial, Rectangular) and if not, the particular profile has to be typed, usually in terms of class marks, class intervals and frequencies. Properly entered 'populations' are labelled and the name printed out for the student's reference. The subprogram DATA allows samples to be generated from these 'populations'. The sample data is labelled and stored but the student can also type in his own set of data if he so wishes. If he needs to remind himself of the population specifications, or enquire about particular sample values, or the amount of store which is left, he can do this by calling the OUTPUT program. One version of the program allows the teacher to set up populations to which the student cannot have direct access. Samples can be drawn in the usual way, and the other facilities are available. This allows data from specific research projects to be taken from the literature and used in the student problems. An example is the investigation carried out by a Research Unit into the convenience of differently numbered sides for the new 50p piece, which was needed when Britain adopted a decimal currency.

An important part of the statistical laboratory is the calculation facilities which are provided. The student can have arithmetical operations carried out on each element of the samples he specifies, or have them operate on corresponding elements between pairs of samples. The sample data can also be summarized as statistics of central tendency, dispersion, order or class, and if these values are themselves set up as labelled lists they can be recalled for calculations to be carried out on them. Thus it is easy for the student to build up his own calculating programs. The package also allows usual numerical calculation and for arrays of formulae to be established in a file so that larger scale programs can be used more directly. Further, there is an editing facility which allows sample lists to be merged, to be adjusted in size, or to be deleted from store.

4.2 Some Experiments

Experiments have been carried out to study the effectiveness of the different teaching strategies, the improvements in student performances when a series of problems is worked through and the relation of these performances to other (university examination) measures of achievement. Two experiments in Chemistry will be briefly outlined, but those in Statistics have followed a similar pattern.

In one study 36 students were randomly allocated to 4 treatment groups ((i) Theory - i.e. teaching of the knowledge of the Chemistry related to the Chemistry experiment - and Planning: (ii) Theory only: (iii) Teaching on Planning only: (iv) Neither Theory nor Planning) where students not having teaching by computer used the normal laboratory work-sheets. The experiment followed analysis of variance design and post-tests were given on theoretical knowledge, planning a similar experiment, and planning an experiment set in a different area of Physical Chemistry. Briefly, both 'planning' and 'theory' showed significant effects, with these groups performing better than the work-sheet groups on the respective tests. However, the interaction between 'theory' and 'planning' did not reach significance, and differences between groups were much reduced on the 'planning a different experiment' test.

A second experiment compared the relative merits of the 'strand' and 'file' systems. It was thought that initially students would benefit from close direction (strand system) but as they become more experienced they should form a plan themselves and collect the information they need to help them. This is best accomplished through a file system (information and planning files). Eventually these help facilities should themselves be faded (information files only). Students were randomly allocated to three such treatment groups with the same material being used in each treatment, but differently organised on the file or strand systems. In order not to penalise the 'information files only' group, a check file was put in and this led to further remedial teaching where necessary. The results were as expected for these new groups of students. The directed (strand) method resulted in significantly higher post-test scores than the file system, which also required longer time. The on-line working gave useful descriptive information on patterns and types of error, and performances on the check files showed the 'information file only' to be the inferior treatment.

Conclusion

Clearly much work has to be done before teaching systems which are adaptive in the sense defined in the earlier sections, can be fully implemented for teaching tasks of this complexity. No author language system presently in use has sufficient facilities or seems suitable in structure. Further, more detailed and careful indentification of knowledge structures and operations, and more experimentation to determine means-ends guidance rules are also essential. However I believe that, in the long term, considerable benefits will come from having the computer take a more intelligent role in the on-line management of learning, and this line of research should be pursued strongly.

These developments might well form an important part of institutionalising the teaching activities of the computer, and for a country with the limited financial resources of Britain a co-ordinated National Effort is needed. This must take into account and enjoy the benefits of related work in other countries. We need a continual exchange of views, which is why we are very properly meeting here today.

References

ABRAMSON, T., WEINER, M., MALKIN, S., and HAVELL, J. 1970. Evaluation of New York City project in elementary arithmetic. Office of Ed. Research. New York.

ATKINSON, R.C., and PAULSON, J.A. 1970. An approach to the psychology of instruction. Technical Report No. 157. Institute for Mathematical Studies in the Social Sciences. Stanford University.

CLOWES, M. 1971. Knowledge systems: How is it possible to acquire them? Paper presented at Conference on AI/CAI. University of Keele. Organised by Science Research Council, and Social Science Research Council, State House, London WC1.

LAUBSCH, J.H. 1969. An adaptive teaching system for optimal item allocation. Technical Report 151. Institute for Mathematical Studies in the Social Sciences. Stanford University.

OETTINGER, G. 1968. Run computer run: The mythology of education innovation. Harvard University Press.

RUBINOFF, M. 1970. In SCHEEPMACHER (Ed.) Planning for effective utilisation of technology in education systems. IFIP Conference on Computer Education. 3; 29-32. IFIP

SMALLWOOD, R.D. 1962. A decision structure for teaching machines. MIT press.

SUPPES, P. 1967. Some theoretical models for mathematics learning. J.Res.and Dev. in Education 1; 5-22.

VAN DER VEER, G. 1970. In SCHEEPMACHER (ed.) Mathematical earning models as tools for computer assisted instruction. IFIP Conference on Computer Education 1; 66-70.

WOODS, Pat, HARTLEY, J.R., LOVELL, K., and SLEEMAN, D.H. 1969. The teaching of arithmetic using a computer based system. In A.P. Mann and C.R. Brunstrom (Eds.) Aspects of Educational Technology 3; 378-385. Pitman.

APPROACHES to the DESIGN and PRODUCTION
of
CAI SEQUENCES

J.F. Hart, R.L. Newkirk
J. Chapman, K. Judd, H. Mackenzie

Department of Computer Science
The University of Western Ontario

APPROACHES TO THE DESIGN & PRODUCTION OF C.A.I. SEQUENCES

SESSION 8A
ROOM/SALLE 148

By/Par: J.F. Hart, R.L. Newkirk, J. Chapman, K. Judd, H. Mackenzie,
University of Western Ontario.

The C.A.I. group at The University of Western Ontario has been looking at the problem of sequence development in the University environment. This practical problem, which seems to constitute a formidable obstacle to any future progress, consists of a number of related factors. Among others we recognize as severe impediments the vast amount of time required to write and test sequences, the inadequacy of methods of quality control, the difficulty of involving psychologists alongside physical scientists and mathematicians, and the lack of a suitable medium of exchange between sequence writers and curriculum planners.

Taken together these problems appear to be a more formidable opposition to the development of tutorial C.A.I. programs than a lack of research in this type of educational technology: the research is in advance of the ability to carry out the development programs.

In this paper we present the first results of a pilot project for the design of quality sequences. An experiment is described in which a group of students and a group of curriculum designers working in parallel developed a course on Elementary Statistics. The results were of two kinds. On the one hand, the experiment indicates the structure of an organization for the design of sequences which in part solves the current difficulties of C.A.I. development. Also, from investigations arising in connection with the joint working sessions of the sequence-writing and curriculum groups, some new directions are indicated including a method of global sequence ordering complementary to the local ordering methods associated with the matrices of Davies.

SUR LA CONCEPTION ET LA RÉALISATION DE SÉQUENCES POUR EAO
(L'ENSEIGNEMENT À L'AIDE DE L'ORDINATEUR)

Le groupe s'intéressant à l'enseignement à l'aide d'ordinateurs à l'Université de Western Ontario, étudie le problème du développement de séquences dans un milieu universitaire. Ce problème pratique, qui semble constituer un obstacle formidable à tout progrès d'avenir, comporte un nombre de facteurs. Parmi d'autres, nous reconnaissons comme étant des empêchements sérieux, la quantité énorme de temps nécessaire pour écrire et vérifier les séquences, l'insuffisance des méthodes de contrôle de la qualité, la difficulté à amener des psychologues à travailler avec des physiciens et des mathématiciens et la manque de moyens appropriés d'échanges de vue entre les auteurs de séquences et les planificateurs de programme.

Si on considère tous ces problèmes, ils semblent poser un plus grand obstacle au développement des programmes de l'EAO que la manque de recherches dans ce domaine; la recherche est en avance sur les possibilités de mettre à exécution les programmes de développement.

Dans notre communication, nous présentons les premiers résultats d'une étude pilote pour l'élaboration de séquences de qualité. Nous décrivons une expérience dans laquelle un groupe d'étudiants, travaillant en parallèle avec un groupe préparant les programmes, a développé un cours de statistiques élémentaires. On a constaté deux sortes de résultats. D'une part, l'expérience nous porte vers la structure d'un organisme élaborant des séquences, ce qui résoudrait, en partie, les difficultés actuelles du développement de l'EAO. De plus, des résultats de recherches qui ont été faites par suite de sessions de travail en commun, montrent de nouvelles orientations parmi lesquelles une méthode globale de classification des séquences complémentant les méthodes de classification locales associées aux matrices de Davies.

1. INTRODUCTION

The research and development work in CAI at Western from the beginning has had as one of its goals, the production of lesson sequences which could be used for Computer Science courses.

With the gift of the coursewriter program from OISE (initially called FOCAL/CAN) the group was able to begin experiments in sequence design without a long period of language development. The first projects were done by graduate students who wrote sequences as part of the Masters thesis. The students accepted the idea, at least at the outset, that the writing of a large tutorial exercise was an acceptable part of research into CAI systems. Everyone in the field knows now the high cost of sequence writing, a fact which the graduate students soon recognized. But while they were still debating the issue and while we had some funds for summer students, we managed to write over 50 hours of sequences.

We soon saw, however, that if we were to continue to develop sequences for use in the department and at the same time isolate less demanding projects for students, we would have to consider the problem of sequence design and production in its own right. Several unfortunate experiments had shown that sequence writing operations could fail, either because of insufficient drive on the part of the sequence writers or because the sequences (once produced) had little merit. Our ultimate objective became the production of sequences having an average lifetime, preferably in a commercial context, of three to five years, rather than sequences for experimental use.

Our examination lead to the broad principles which we consider essential in a productive program. In general, we began to see the need for interdisciplinary co-operation which would involve specialists in course content, in pedagogical theory and in CAI technology, as well as a degree of student participation to infuse an acceptable dialect for the particular sets of sequences. At the same time we recognized that universities could not continue in sequence production if costs continued to be high, if the development time was prohibitively long, and if the sequences were not able to pass certain criteria for quality control. It was an attempt to satisfy these requirements which lead to the experiment to be described.

Student participation in the production of CAI sequences may be seen in the larger perspective of the development of a new educational technology. Programmed instructional sequences, whether for computing or print, must be written in such a way that they are appropriate to the group which is to use them. Student sequence writers may contribute to the appropriateness of sequences by the vocabulary they use and the manner in which they structure the course (Mager, 1961). However, their inexperience and unfamiliarity with the latest research may result in sequences not meeting norms of professional competence and for this, it is necessary to have the consultation of experts. On the other hand, it is not unusual to find that sequences written by someone knowledgeable in a subject turn out to be completely unusable. The adaptation of sequences to the domains of both user and expert increases the time investment required, but may also result in greatly increased effectiveness. This gains in importance as CAI lessons begin to be judged on their own merits, rather than as comparative improvements over previous teaching methods.

This paper describes an experiment in which students were given the job of designing and writing sequences for elementary statistics. The purpose of the experiment was to see if they could do so in a way that offered the possibility of quality control during the writing process. If so, a new approach to CAI production [Hart et al, 1970], might be pursued. This approach requires that there exist alongside the sequence writing group, a curriculum group with the required expertise in course content, pedagogy, and psychology. Essentially, the experiment consisted of the writing of three tutorial lessons for Introductory Statistics, using the basic facilities of a teletype terminal. The experiment was not concerned with the best hardware and software environment for design and production. It was cast on the wider perspective of considering organizational structure of sequence-writing groups in terms of the general requirements for production of quality sequences. The question was how to construct a CAI industry, making use of relatively untrained personnel with, however, some experience in the content to be programmed. Three students (Chapman, Judd, and Mackenzie) were given the task of designing and producing sequences. The responsibility for writing the sequence was theirs, but they were asked to call upon the consulting group in various aspects of the work.

2. The Organization of Sequence-Writing Groups

The implementation of methods to establish quality control is basically an aspect of organizational structure. Starting with the organization chart of the agency engaged in production, studies are made of methods of improving the quality of the output.

The problem is complex, but it can be seen in its essentials by looking at the way CAI has been developed. Until now, sequences have often been written by a teacher or by students under the direction of a teacher. The relation of the student in this situation is that of assistant, comparable to that of a computer program coder to a systems analyst. In large projects, this structure can become strongly hierarchized, the apex of the decision tree being a curriculum group which in effect, controls all aspects of the work. If students are not to become mere clerical assistants, it is required to reconsider this way of operating from the viewpoint of the decision structure.

In the study of Argyris [1964], the value of a particular decision structure is measured by the kinds of results which are expected to be obtained. A hierarchized structure is effective in certain situations such as when time is short or when decisions are of such a kind that they may be delegated to a leader with the consent of the group. On the other hand, a horizontal or democratic structure is applicable to situations such as those associated with the development of a new product or long-term planning. In the management of novelty, which is the case of CAI, the attribution of power is along functional lines. The ability to originate actions is given to an individual or a group in accordance with an ability to contribute to the solution of a problem. Argyris summarizes the guiding principle of decision structure as follows:

> We may conclude that organizations (of the future) will tend to vary the structure that they use according to the kinds of decisions that must be made. If one asks the individual in the organization of the future to see the company organizational chart, he will be asked "For what type of decision?"

The type of structure which we require for CAI is one in which a student group is given power of decision in the writing of sequences, but which would also permit researchers to influence the design. The proposed organizational structure consisted of two organically distinct working groups with separate functions. It was considered to be essential that there should be no relation of dominance but that the groups should work in parallel, one to apply the general or formal principles of CAI design and the other to actually prepare lessons.

The functions of the groups were not rigorously defined since the operations of the groups were expected to change, depending on deeper understanding of different aspects of sequence design. At this initial stage, work was divided from an operational standpoint between a curriculum group concerned with concept ordering and a sequence-writing group, concerned with the programming of sequences. The focus and language of the groups were different. The sequence group is concerned with a dialect appropriate to the age, culture and intellectual background of the students, whereas the curriculum group is concerned with a metalanguage of concept theories and pedagogical schemes. This difference in working language is closely associated with the relationship to specific techniques. The sequence writer must be versed in the technology of interactive computing, whereas members of the curriculum group are more likely to be concerned with techniques and theories in the psychology of learning.

Assuming the existence of the two groups, a schematic must be designed so that the findings of both groups may be communicated to the common task. It must be simple and immediate and sufficiently permanent to allow the two groups to work at essentially different rhythms. A satisfactory link, described later on in the paper, adopted a technique similar to the clothboard in which individual concepts were represented by a moveable set of cards.

3. Strategies of Conceptual Ordering

The ability of the curriculum group to contribute to sequence design, can only be on the basis of certain practical suggestions or strategies which it can contribute to the ongoing process of sequence writing. Such strategies could be at the specific level of knowledge about a given field, but in order to be subject-independent and to act as overall engineering principles they must relate to organization of concepts in general, as well as to some specific field.

General strategies of concept ordering may be divided into two kinds: those relating to microstructure or the association of concepts in pairs, and those relating to macrostructure of the global organization of the concept domain. In applying strategies of the first kind, it is required to isolate the key concepts or rules of the material to be taught and to examine the relations of association or discrimination which exist between these key concepts. Eventually the concepts become grouped in clusters by techniques such as the matrices of Davies [Thomas et al, 1963].

The set of strategies based upon macrostructure are the outgrowth of research conducted by Denny [1970], in the study of concept formation. In this analysis, the elements are, as in Davies, the individual key concepts, but examined according to the unifying aspects of a domain of knowledge considered as a whole. The analysis is undertaken in the light of two closely related principles in General Systems. The first principle is that of system unity; whether in fact a certain type of knowledge exists operationally in the mind as a unified whole. The second principle is that of system linkage, whether there exists a connecting link which brings together broad regions of a field and supplies an operational unity. The first principle helps to isolate separate or nearly separate subsystems, whereas the second emphasizes the need to search out unifying schema or formulae.

From these principles, the following strategies were provided by Denny, as a type of operational working hypothesis:

1) Identify the conceptually distinct areas within the subject.
2) Link concepts by means of their relations to more general concepts.
3) Convey to the user at the outset the overarching generalization (key general notion) of each content area.
4) Combine secondary notions into an overall organization determined by the integral notion of the field.

It might appear that the Davies and Denny strategies are essentially the same. The difference is that the Davies approach is from the inside out or from pair association toward the recognition of clusters in the larger organization whereas the Denny approach is from the outside in and consequently from the boundaries of the subject matter toward a definition of broad regional divisions and general linkages. The two approaches should thus be complimentary and mutually correcting. In the work of the pilot group, only the Denny strategies were actually applied.

4. Formation of the Groups

The curriculum group was readily established since in effect it existed already in the form of an association of university researchers interested in educational technology. The group included members with backgrounds in media, psychology of learning, and concept formation. The sequence group was selected from a list of students who applied for summer work. The members of this group had backgrounds in social science, general studies and computer science.

Prior to the selection of the students, it had been decided that the field to be used for the experiment would be Introductory Statistics. Knowledge of this field was required by members of the sequence-writing group.

The division into two groups was not rigid - certain individuals might find that their occupations connected them more closely to one or the other, particularly if they were teachers.

The two groups adopted quite different methods of working. The project was the students' continuous occupation for four months, whereas it was only one of a variety of university occupations for members of the curriculum group. Each group decided on its own procedures. Generally speaking the time limits for the completion of parts of the project, were primarily determined by the requirements of the sequence group.

5. Communication Between Groups

Since the curriculum and sequence-writing groups worked independently, an important consideration was the development of an effective means of communication between the two groups. A graphical display of the subject matter organization was selected as the primary means of communication. As the project was housed in a separate room, the walls were used as a display surface for coloured cards. These cards contained a breakdown; first, into the two main areas of statistics (descriptive and inferential); then within each of these areas, into topics (e.g. regression analysis, analysis of variance) and finally within the topics, into component items. The topics were arranged in a proposed order of presentation, and each topic was described by prescribing the items which were "entering behaviour", "intermediate lesson content", and "terminal behaviour" for that section.

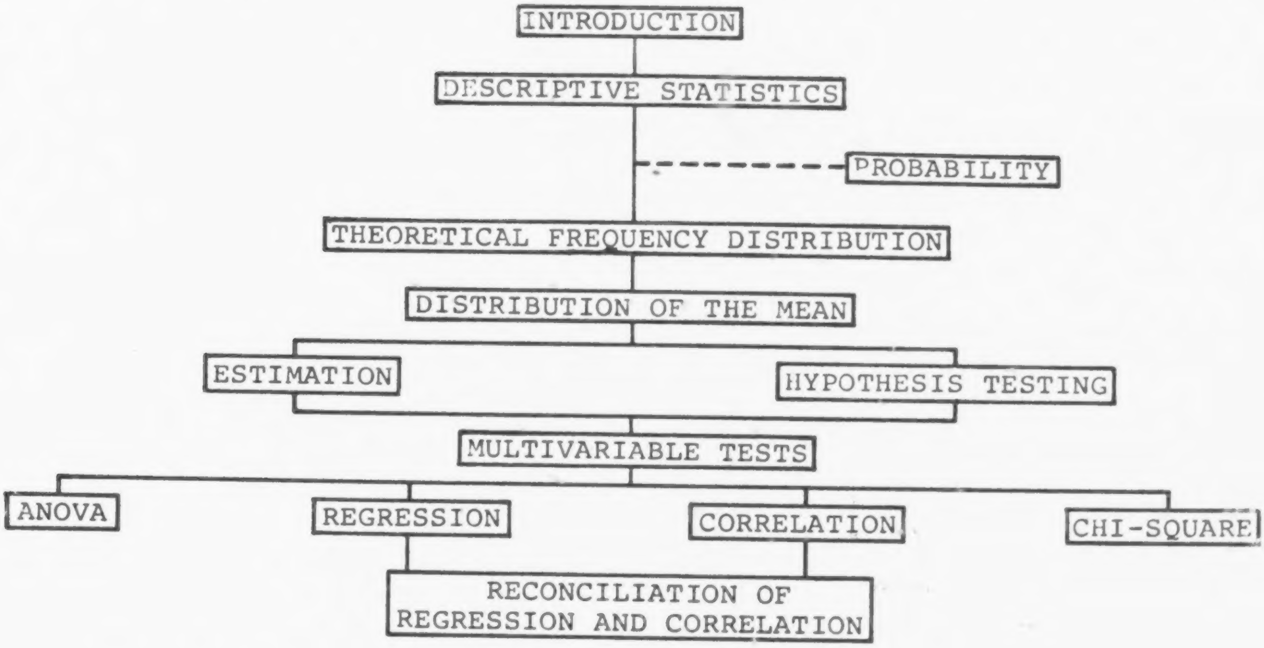
This method of presentation allowed for effective communication between the two groups. The representation involved only the concepts of statistics. The easy movement of cards contributed to a flexible approach to the organization, both initially and during later revision. By means of the cards and their schematic relations, all the relevant information could be conveyed to the curriculum group at meetings convened for the purpose. The graphical representation also had merit in that it made clear visually structures of a type that are used in both the systems approach and a conceptual organization of material and was particularly useful to the curriculum group.

Eventually, a computerized graphical display will provide a better method. The design of a suitable system will be facilitated by this initial experience involving cards.

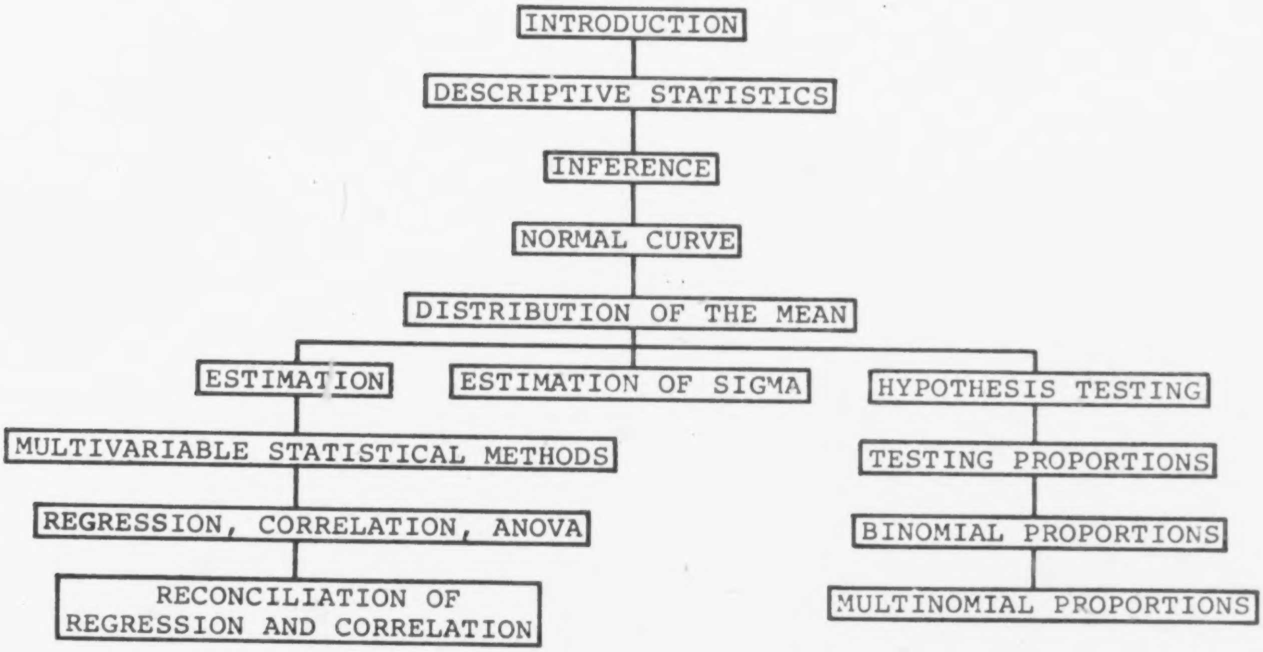
6. Division and Ordering of Concepts

Before any lessons could be written, it was necessary to locate them within the field of statistics by doing a complete analysis of the subject and arranging it as an entire course in some detail. This was done in two steps by the sequence group. First, an exhaustive summary was made of the topics which were felt to comprise the subject "Introductory Statistics", and the component items in each of these topics. This was done primarily by reference to traditional textbooks, course summaries, etc. Second, the traditional organizations of these topics into complete courses was examined critcally and an organization was derived from this and from a study of prerequisites and relationships among the topics. In this organization, the group was influenced by their experiences in the study and use of statistics, particularly in formal university courses. They endeavoured to arrive at what seemed a "best" organization from a student's point of view. This initial organization, recognized to have some weaknesses, formed a basis for discussion with the curriculum group, who were in turn guided by the principles of conceptual ordering cited. Their suggestions resulted in a number of changes, which may be illustrated by comparing the initial and revised scheme in Figure 1.

INSERT FIGURE 1 HERE


INTRODUCTION
DESCRIPTIVE STATISTICS
PROBABILITY
THEORETICAL FREQUENCY DISTRIBUTION
DISTRIBUTION OF THE MEAN
ESTIMATION
HYPOTHESIS TESTING
MULTIVARIABLE TESTS
ANOVA
REGRESSION
CORRELATION
CHI-SQUARE
RECONCILIATION OF REGRESSION AND CORRELATION

INITIAL COURSE STRUCTURE


INTRODUCTION
DESCRIPTIVE STATISTICS
INFERENCE
NORMAL CURVE
DISTRIBUTION OF THE MEAN
ESTIMATION
ESTIMATION OF SIGMA
HYPOTHESIS TESTING
MULTIVARIABLE STATISTICAL METHODS
REGRESSION, CORRELATION, ANOVA
RECONCILIATION OF REGRESSION AND CORRELATION
TESTING PROPORTIONS
BINOMIAL PROPORTIONS
MULTINOMIAL PROPORTIONS

REVISED COURSE STRUCTURE

FIGURE 1. STRUCTURE OF ELEMENTARY STATISTICS COURSE BEFORE AND AFTER APPLICATION OF PRINCIPLES OF CONCEPTUAL ORDERING.

On the basis of the first principle of conceptual ordering - identify conceptually distinct areas - sections of inference using raw scores as data, and sections using frequency classes, were clearly separated. In conjunction with this, separate sections on binomial and multinomial propertions were introduced. The second principle resulted in the addition of a section designed to tie together the concepts of regression, correlation, and analysis of variance by referring to their relationship to a general equation relating dependent to independent variables. In order to convey the key notion of each area to the student at the outset (principle three), careful introductions were included for all lessons. The fourth principle, that secondary notions should be integrated into an overall organization determined by the integral notions of the field, was the basis for the most significant changes in the course structure. In the initial proposals, sections on probability and on theoretical frequency distributions had appeared as separate units, near the beginning of the course Although these are not central to the development of the field of statistics, they are traditionally taught separately at the beginning of a course, so that the student has that knowledge later on when it is required for the understanding of more central concepts. However, it is difficult for the teacher to explain, or the student to see, the real purpose of the material when it is introduced at this time. Therefore, the separate sections on probability and theoretical frequency distributions were deleted. Instead, relevant items from these topics were introduced as needed. For instance, random variables, events and sample space were part of the first section on inference, one of the aims of which was to introduce probabalistic statements as distinct from descriptive statements. The normal curve was taught as it was required, just before the section on the distribution of the mean. The binomial distribution was introduced when it was necessary in a section on "Testing Proportions." These changes can be seen by comparing the initial and revised scheme.

During the revision process, several informal meetings were held between the curriculum and sequence writing groups. The final organization of subject matter was accepted and endorsed by both groups. Many of the changes to the course structure occurred in sections about which the sequence writing group had doubts, but lacked the theoretical background to realize exactly what changes should be made. The work of the curriculum group provided the justification for improvements in these areas; the graphical presentation provided the means for these theoretical notions to be conveyed to the sequence-writing group, who could then act upon them.

At this time, the sequence-writing group began to work on individual lessons. Detailed plans for the lesson structure were drawn up and evaluated critically within the group, before the lessons were actually written. Writing, coding and testing proceeded in the normal manner.

7. Project Evaluation by Curriculum Group

The organization of statistics which was achieved was felt to be very satisfactory from the point of view of the curriculum group.

The merit of the principles of conceptual ordering as a practical guide to course structure will only be fully determined when sequences for the whole course are completed, and the effectiveness of this organization can be compared to that of a traditionally organized course. However, the curriculum group was satisfied that the current organization reveals the overall structure of the subject matter clearly, and that those students involved in the project felt it would have been helpful had such an organization been used in their statistics course.

The curriculum group was satisfied that the students can successfully play a major role in the development of CAI material, allowing the sequence group considerable freedom and responsibility for the material produced. This model is believed to be a major contributing factor to an ongoing high level of motivation and interest in the project. (Previous experience has shown that sequence writers often become bored and unenthusiastic during the more tedious parts of the project, such as coding). This enthusiasm and the involvement of the sequence group in all phases of lesson design and writing, have also resulted in an excellent quality of work at all stages. The students showed themselves to indeed be capable of applying standards of quality control to their work, supported at the three main levels - course structure, lesson structure, and lesson presentation - by the evaluations and opinions of the curriculum group, CAI technologists, and subjects who tested the lessons.

The use of students as sequence writers, although not fully consistent with the philosophy expressed by Mager (a fully consistent application would be a totally unstructured course) at least recognizes that the optimal path through a subject area for a student may differ markedly from that of a teacher.

8. Project Evaluation by Sequence-Writing Group

The students felt from the outset that a pragmatic approach should be taken to the ordering process. This was expressed in the sequences written by introduction of concepts when needed by the student rather than when convenient for the teacher. For example, standardization of raw data was taught with the normal curve, when it can be applied meaningfully, rather than after measures of central tendency, when it cannot readily be applied.

As students, the sequence group were more aware of the areas of conceptual difficulty in the subject than an expert in the field might be. They felt that in most introductory textbooks and courses, too much emphasis is placed on formulae and not enough on explaining concepts and developing a feel for the subject. They saw tutorial CAI as an ideal method for teaching and verifying these concepts.

They were particularly aware of the level at which the material was to be taught. Since this was an introductory statistics course, formal theory was avoided as much as possible. Concepts were developed in a clear, intuitive manner rather than using formal proofs which may be much more formidable to the student than the result itself. Elegance was sacrificed to clarity if the two were in conflict.

A feeling that is often easy to "miss the forest because of the trees" in a statistics course lead the sequence group to make special efforts to tie the blocks of the course together and make extensive use of introductions and summaries. This combined well with the curriculum group's second and third principles to produce many of the final modifications to the overall course structure.

The sequence-writing group found that to a surprising degree, their confusions and skepticisms were fairly reliable indicators of weaknesses in concept ordering. It had been one of the assumptions of the project, which appears to have borne out that the students would be less attached to traditional methods of presentation than the instructors would be, and thus able to make more constructive use of the suggestions of the concept theorists.

Working as a group, the students in the sequence group were able to set a high standard of quality control, by maintaining a constant check through discussion on each others' work. Although the duration of the project did not permit it, it was thought that it would be desirable, once writers had choses specific lessons for which to write sequences, that a structural breakdown in the same fashion as the overall breakdown and involvement of the curriculum group at this stage as well might have been desirable.

9. General Conclusions

As the curriculum group had observed, the project appeared to verify, as much as possible in the short time available, the thesis that students could carry out work of high quality in a self-actuating manner, provided that the work was organized around two groups working in parallel, one concerned with ordering strategies and the other with lesson writing. The ability of the students to work on their own, and the mechanism for implicating curriculum theorists were the solvent features. This mode of operation is of great interest in that it permits students an active and indeed central role in the educational process. Following this procedure, it would seem reasonable to establish courses in which the principle work of the course, (at the same time a learning and teaching activity of the part of the student) would consist of the preparation of CAI materials.

Some aspects of the project deserve further investigation. For instance, the utility of the strategies of concept ordering indicate the possibility of an increased role for systems theory in the design of a statistics course.

But the main result of the project is a new approach to design of CAI sequences. Seen in relation to the problem of the formation of technical ensembles, the project helps to discriminate between the role of the individual and the role of the group in the development of educational technology. The contributions and the needs of the individual have always been stressed as an argument for programmed instruction. But what does not seem to have been noticed, is that in order to enhance the value of the information provided to the individual, there must be a correlative increase of the involvement of various groups.

The situation may be illuminated by comparison with an educational technology based on printing. Education based upon the book is individualistic, not only because the manuscript multiplies into a number of copies to serve individual readers, but because its main technique, the printed text, tends to be the work of one person or of one mind. It seems, however, that CAI does not follow this model. There is a similar multiplication of copies but CAI should not be a mode of preservation of the thought of the single teacher, which the book does well, but the concretized influence of teacher, student and concept theorist. In a word, whereas the book is based on an artisanal operation, CAI is based on a technology of the group.

If this thesis is accepted, there are two aspects which we shall present by way of conclusion. The first has to do with the fact that until now course-writing languages have been designed primarily to serve the individual teacher. By what has been said above, this is in effect to design CAI technology on an artisanal model. By contrast, our study raises the question of how to specify such a language as the working medium of a group.

A second point concerns the role of the individual teacher in CAI. If CAI is group-centered, the teacher must become involved as part of a group and in collaboration with curriculum and student groups. Thus the individual teacher must search for a new type of engagement in the educational process. This does not imply that the roles of the teacher in any one of the traditional senses must disappear, but that they must be expanded to include the new collective activities. Just as the student loses his narrow dimension of learner, so the teacher loses his narrow dimension of professor. This could have salutary benefits for the renovation of the educational system as a whole, because the change cannot come about without the consent of the principal groups concerned and the new technology is therefore free to grow up naturally within a particular social environment.

REFERENCES

Argyris, C.: Integrating the individual and the organization. John Wiley and Sons, New York, 1964.

Denny, P.J.: Principles of the conceptual ordering of instructional materials. Research Bulletin No. 206, Department of Psychology, University of Western Ontario, October, 1971.

Hart, J.F., I.P. Suttie, and A.G. Slemon: Establishment of technical ensembles for sequence design and production, 1970.

Mager, R.F.: On the sequencing of instructional content. Psychological Reports, No. 9, (1961), pp. 405-413.

Thomas, C., I.K. Davies, D. Openshaw, and J.B. Bird: Programmed Learning in Perspective. Lawson Technical Products Limited, London, 1963.

LEARNING TO USE AN INTERACTIVE COMPUTING SYSTEM

Mrs. Beverley Colpitts

Department of Computer Science
The University of Western Ontario

LEARNING TO USE AN INTERACTIVE COMPUTING SYSTEM

SESSION 8B
ROOM/SALLE 148

By/Par: Mrs. Beverley Colpitts,
Department of Computer Science,
University of Western Ontario,

Successful use of an interactive computing system demands a basic knowledge of the hardware and software which together comprise the on-line system. Misuse can be costly in terms of time and money. At the University of Western Ontario, many undergraduate Computer Science students are required to use a Digital Equipment Corporation PDP-10 computing system. Each year, when an influx of new students occurs, the problem of communicating sufficient information to assure competent usage presents itself.

During the summer of 1971 a team effort resulted in a teaching package consisting of two video-taped lectures, nine C.A.I. lessons and several pages of prepared notes. Its aim is to provide an opportunity for self-paced, independent learning by any student requiring information about the on-line computing system. The lectures serve to introduce some basic concepts in a graphical way, while the prepared notes provide the students with a permanent copy of detailed information. The C.A.I. lessons, the most vital element in the package, enforce concepts in the lectures, introduce new concepts and in some senses simulate the on-line system for the student.

During the Fall Term approximately one hundred students in Year II of a BSc program in Computer Science used the learning package. The video-taped lectures were shown to the students in groups. Each student reserved a carrel in the C.A.I. laboratory for two hours per week for the purpose of running lessons. The laboratory contains ten carrels, each equipped with a quiet terminal with a phone-line connection to the central computer, and a remote access slide projector and rear-projection screen. Students were free to run as many lessons as time permitted in a session at the terminals. Typed material was available on request from the laboratory supervisor.

Using the on-line computing system to teach about itself has a number of advantages. The one-to-one nature of the situation requires that each student become an active participant in the learning process. As the student runs the C.A.I. lessons, his familiarity with an on-line terminal itself increases and consequently his confidence grows. He is exposed directly to the behaviour of the on-line system, growing accustomed to such features as variation in response time. In the advanced lessons, the opportunity to type commands to the teaching program, which functions in a simulation mode, allows the student to practise forming commands under the protection of the C.A.I. lesson.

COMMENT APPRENDRE À CONVERSER AVEC UN ORDINATEUR

Pour bien converser avec un ordinateur, il faut savoir ce que c'est qu'un ordinateur et en quoi consiste les programmes. Un mauvais usage de l'ordinateur peut être très coûteux et faire perdre beaucoup de temps. A l'Université de Western Ontario, de nombreux étudiants en informatique doivent se servir d'un ordinateur PDP-10. Chaque année, nous devons résoudre le problème consistant à nous assurer que les étudiants sauront se servir de cette installation.

Au cours de l'été 1971, une équipe a enseigné comment se servir de cette installation à l'aide de deux cours enregistrés sur bandes magnétiques audio-visuelles, de neuf leçons présentées par ordinateur et de quelques pages de notes préparées. Le but de cette équipe est de faire en sorte que tout étudiant puisse s'instruire indépendamment et à sa propre vitesse en faisant appel à la liaison dont il dispose avec l'ordinateur. Les cours présentent quelques notions de base sous forme graphique et les notes fournissent des renseignements détaillés. Les leçons, élément le plus intéressant, renforcent le cours, présentent de nouveaux concepts et, en quelque sorte, simulent la liaison avec l'ordinateur.

Durant l'automne, environ cent étudiants en deuxième année du premier cycle en informatique ont acquis de l'expérience avec cette installation. Les conférences audio-visuelles enregistrées ont été données devant des groupes d'étudiants. Chaque

A geography question appears on the cathode ray tube; after reference to a map and some deliberation, the student types an answer and immediately receives confirmation that he is indeed correct. A systems programmer, tracking down a programming bug, types a command to suspend execution at a particular point in the troublesome routine; after interrogating the system for certain variable values, he commands the system to make a change in a variable and to resume execution of the routine; output indicates the error has been pinpointed. An engineer indicates with a light pen where he wishes to make a structural change in a bridge design; within seconds the computer informs him of the effects of these changes. These applications illustrate the interactive nature of on-line computer usage. The user and system together form a powerful man-machine partnership. The system requires decisions and input from the user at various stages during a program's execution. The user requires the output produced by the system in response to his earlier commands as a basis for further decisions. The versatility of this partnership is a consequence of the direct link between the user and computer system made possible in the on-line environment.

The cost which the user must pay for interaction is increased responsibility during the development and debugging of programs. Unlike a batch-processing system, where the user loses contact with his program from the time the deck is submitted for running until the output is picked up, the on-line system does not isolate the user from his program during any phase of its processing. The user sits at a terminal which is connected to the computer and interacts with the system. He may enter his program by typing at the terminal. He must then direct the system to compile, load and execute his program. At any time he may interrupt what is happening to make changes in his program or to issue new commands to the system. Thus successful use of an interactive computing system demands a basic knowledge of the hardware and software which together comprise the on-line system. Misuse can be costly in terms of time and money.

At the University of Western Ontario, several undergraduate Computer Science courses require students to use a Digital Equipment Corporation PDP-10 time-sharing system. Each year, when an influx of new students occurs, the problem of communicating sufficient information to assure competent usage presents itself. Unfortunately, the required information was not readily available from a single source. Instead, pertinent information was scattered throughout a number of computer manuals, which are intended primarily as reference manuals rather than as teaching aids. Nowhere in the manuals was a suitable overview presented. Initially the course instructors concerned tried to communicate the necessary information by lecturing independently to their classes, with great duplication of effort. A review of the situation, prompted by feedback from the students, clearly indicated the need for a co-ordinated teaching package.

Planning An Integrated Instructional System

In Spring, 1971, a team of six members assumed the responsibility for developing a teaching package. A faculty member, two course assistants and three undergraduate students comprised the group. The faculty member and course assistants contributed an interest in teaching, and experience in presenting information about the on-line system in the conventional lecture manner. The students brought to the project enthusiasm, recent learning experiences, and few prepossessions about teaching.

The overall goal of the teaching package was to provide an opportunity for self-paced, independent learning about the on-line system such that upon completion of the package a student would be sufficiently competent and confident to enter, debug and test a Fortran program on-line. It was unanimously agreed that the package must do more than teach the student a recipe to follow step by step. It had to present a co-ordinated picture of the computer system, stressing the role of system components and the purpose of each step in the processing of a program. With this information, a student should be better equipped to adapt to a situation differing in some respects from examples presented in the package.

With the overall goal and a more detailed description of the desired terminal behaviour in mind, the group tackled the identification of concepts and related system commands which must be presented in the package. The undergraduate students were of particular value during this planning phase as they were able to point out where they had experienced difficulties and misunderstandings.

The next phase involved a review of available instructional media and the roles they could play in this instructional system. Video-tapes, films, slides, reading material and CAI were considered. Subsets of the concepts and details to be taught were discussed in relation to possible methods of presentation. One fact became very clear - no one medium could satisfactorily communicate all the information. Instead, an integrated instructional system seemed desirable, with each medium being used where it seemed most advantageous.

Some of the material to be presented was better suited to a pictorial rather than a verbal presentation. In particular some topics lent themselves very well to animation. This seemed an ideal application for video-tapes since part of their potential is the ability to present information in a manner which would be difficult, if not impossible, to use in a live lecture. The philosophy was thus to use the video-tapes to present "television lectures" rather than simply "televised lectures".

The information to be presented also included detailed PDP-10 commands of which the student must have a permanent copy. Reading material seemed appropriate here.

Last, but not least, it seemed that CAI could play a vital role. Interaction was the key. CAI could make the student an active participant in the learning process, by asking him to demonstrate his understanding of one concept before another was presented. As a bonus, students would gain experience with the actual on-line system and terminals about which they were to learn and would gain confidence in a new environment. It was felt that slides should be integrated with the lessons.

It was not enough just to realize that each of the media mentionned had a role to play in the teaching package. Careful consideration had to be given to the degree to which one part of the presentation should depend on another and to the nature of this dependence. While modularity could increase the usefulness of the package, it was of utmost importance that the end result should not be just a collection of devices but rather an integrated instructional system.

The final package is comprised of two video-tapes, nine CAI lessons and several typed handouts. The video-taped lectures serve to introduce basic concepts in a graphical way, while the prepared notes provide a permanent copy of detailed information. The CAI lessons, the most vital element in the package, enforce concepts presented in the lectures, introduce new concepts, and in some sense simulate the on-line system for the student.

The CAI Component

The nine CAI lessons included in the teaching package may be divided into two groups. Seven of these lessons deal with a number of system components and the steps in processing a Fortran program. The remaining two lessons deal with a specific system program called TECO which allows on-line creation and editing of files. The lessons are coded in ETL (Educational Technology Language).

Each lesson can easily be run in a one-hour session at a terminal. This constraint on length was imposed because of the difficulty of concentration for longer periods and to enable students to use a spare period between classes to run lessons.

The style of the lessons is informal and friendly, and occassionally humorous. These lessons bear the mark of their enthusiastic authors! For example, the group conceived the idea of personifying certain system components as cartoon characters to better help students remember important characteristics. The monitor was represented as an octopus, with each of his arms busily looking after a user of the time-sharing system. The disk was presented as a filing cabinet character, where the many drawers reminded students that the disk is a sharable device where each user has his own private area (drawer).

The structure of the lessons is generally a spiral one. Concepts are introduced and skimmed over lightly. The same concepts are then presented again in more depth. Finally, the PDP-10 commands related to the concepts are presented. This approach was chosen because the system components are so interrelated that it is difficult to separate one from the others in order to discuss it in detail.

A variety of instructional strategies are employed throughout the lessons. These include drill and practice, tutorial, simulation and inquiry strategies. As an example, let us consider the seven on-line system lessons. The first two lessons serve mainly to enforce concepts introduced in the video-taped lecture. Consequently, they use a blend of drill and practice and tutorial strategies. Lessons three and four expand on these concepts and introduce related concepts - the strategy here is chiefly tutorial. Lesson five provides drill and practice on the use of the PDP-10 commands. Lesson six uses a limited simulation strategy, giving students a chance to practice the step-by-step processing of a Fortran program. Lesson seven is a review lesson which allows the student to choose a topic from a list of available topics, hence operating in an inquiry mode.

All nine lessons make active use of slides. These slides serve to present text if it is rather long to be typed out by the lesson or to present pictorial information.

Production and Testing

The team of six divided into smaller groups during the actual production of the teaching system components. Two members worked on the TECO lessons and typed hand-outs. They co-ordinated their efforts through discussion, but chose to assume separate responsibility for the production of one CAI lesson each. The remaining course assistant and students tackled the more general on-line lessons and notes. They employed a different strategy as all three contributed ideas, questions, and examples to a single lesson. The faculty member acted as a supervisor and resource person, and assumed responsibility for the production of two video-taped lectures. The entire group met regularly to exchange ideas and to report the progress of the subgroups. These meetings were imperative to ensure the co-ordination of the package components.

Time constraints did not permit major testing of the package before the end of the summer. The team spent many hours doing testing and solicited the aid of other department members, but no sizable group of students tested the package before its actual introduction into a number of undergraduate courses.

Application

During the academic year 1971-72, approximately one hundred and seventy-five undergraduate Computer Science students used the learning package. At the outset each student received a guide which clearly outlined the chronological order to be followed in using the video-tapes, typed notes, and CAI lessons.

The introductory video-tape was shown to a class during a regular lecture period. The second video-tape was shown in small groups as they reached the appropriate point in the course. Replays were available at anytime upon request.

Lessons were run in the CAI laboratory. The laboratory contains ten carrels, each equipped with a quiet terminal with a telephone-line connection to the central computer, and a remote access slide projector and rear-view projection screen. Five of the terminals produce hard copy while five do not. A supervisor is present at all times to assist students who encounter equipment or system problems.

Each student reserved a carrel in the laboratory for two hours per week for the purpose of running lessons. He was free to run as many lessons as time permitted in a session at the terminal and was also at liberty to spend more than the two hours per week in the laboratory if a terminal was free. Typed material was available on request from the laboratory supervisor. On the average, students completed the package in three to four weeks.

Results and Conclusions

Student reaction to the teaching package was generally favourable. Some feedback was available informally while the students were involved in the course. An evaluation form was used to solicit more formal criticism some time after the course was completed. Students indicated that in their opinion video-tapes, reading material, and CAI lessons were all valuable components of such a package.

Most comments concerned the CAI lessons. Students agreed that the lessons were interesting and that block size, question difficulty, information density, and lesson length were satisfactory. They felt that the overview could be more strongly emphasized and that additional drill on the PDP-10 commands should be included. As could be expected, some questions were considered to be unclear and a number of minor bugs were found. This feedback and an analysis of the statistics collected for each question will hopefully provide a good basis for making improvements in the lessons.

A majority of the students commented that hard copy of their lesson runs was highly desirable. This was emphasized by the fact that students entering the laboratory selected the terminals without a provision for hard copy only if all the other terminals were already in use. The hard copy assumed importance after the running of a lesson rather than during it - i.e. students did not wish to refer to the listing during the lesson, but wished to study the listing at a later time. This may indicate a weakness in the typed material included in the course, as its role was to provide the student with a permanent copy of detailed information.

Comments and criticism are valuable but "The proof of the pudding is in the eating". At the conclusion of the course, the students were given an assignment to use the PDP-10 on-line system to enter and run a Fortran program, to make specified changes in this program and to run the modified version. The student was then to bring the results of his efforts to either his lecturer or a course assistant for discussion. The students were not instant experts. They made mistakes and spent far more time doing the assignment than an experienced user would. However, they approached the assignment with a degree of confidence and were able to complete it with little help from anyone else. Discussion revealed that they were equipped to understand the consequences of the mistakes they made. It was also interesting to note that these students made the transition from processing Fortran programs on-line to processing assembly language programs on-line very smoothly.

This teaching package seems to fit well into an undergraduate Computer Science course. The attempt to integrate several media was appreciated by the students. Using the on-line system to teach about itself has a number of advantages. The one-to-one nature of the situation requires that each student become an active participant in the learning process. As the student runs the CAI lessons, his familiarity with an on-line terminal itself increases and consequently his confidence grows. He is exposed directly to the on-line system, growing accustomed to such features as variation in response time. In the advanced lessons, the opportunity to type commands to the teaching program allows the student to practice forming commands under the protection of the CAI lesson. Here mistakes are learning experiences without severe penalties in terms of time and money.

REFERENCES

PDP-10 Timesharing Handbook, PDP-10 Handbook Series, Digital Equipment Corporation, Maynard, Massachusetts, 1970.

PDP-10 Reference Handbook, PDP-10 Handbook Series, Digital Equipment Corporation, Maynard, Massachusetts, 1969.

Introductory Manual for ETL, Computer Science Department, The University of Western Ontario, 1972.

MÉTHODE ANALYTIQUE DE FABRICATION DE COURS CAI

SESSION 8C
ROOM/SALLE 148

By/Par: Robert Brien,
Professeur,
Service de l'Informatique.

Plusieurs chercheurs considèrent la construction de cours valables dispensés par ordinateur comme étant un problème fondamental.

La nécessité pour l'auteur de maîtriser un langage de programmation, le temps mis à la construction d'un cours, le manque de stratégies bien définies font que de nombreux pédagogues se privent d'un excellent outil d'enseignement.

Dès l'implantation du Système 1500 en janvier 1970, des membres du Laboratoire de Pédagogie Informatique de Québec se penchèrent sur ce problème. Diverses techniques de construction de cours furent expérimentées dont une méthode globale basée sur des modèles d'enseignement (stratégies) qui permet à l'auteur de contourner les difficultés de la programmation.

Cette première méthode de construction de cours développée et utilisée par les membres du L.P.I. força ces derniers à s'interroger sur les critères de validité d'un cours dispensé par la machine et à redéfinir de façon plus rigoureuse les actions pédagogiques que devrait adopter l'auteur dans l'élaboration de son cours.

Le but de l'exposé est justement d'expliquer comment, à partir de la méthode SPC élaborée en 1970, nous en sommes venus à une nouvelle approche susceptible d'aider l'auteur dans le développement de son cours CAI.

ANALYTICAL METHOD OF CONSTRUCTING CAI COURSES

Many researchers feel that the design of worthwhile courses to be given by computer presents a basic problem.

Because of the necessity for the author to master a programming language, of the time taken in constructing the course and of the lack of well-defined strategies, many teachers do without an excellent instructional tool.

As soon as they received the 1500 System in January 1971, the members of the Computer-aided Teaching Laboratory of Quebec began studying this problem. Various methods of designing courses were tested one of which was a global method based on teaching models (strategies) which allowed the author to get around the difficulties of programming.

This first method of course making, developed and used by the members of the Computer-aided Teaching Laboratory, forced them to question the validity criteria of a machine-given course and to redefine in a more rigorous way the teaching practices that the author should adopt in setting up his course.

The aim of this paper is precisely to explain how, beginning with the SPC method designed in 1970, we have arrived at a new approach capable of helping the author in developing his computer-aided instruction.

DESIGNING PROGRAMMING LANGUAGES FOR COMPUTER-BASED INSTRUCTIONAL SYSTEMS IN RELATION TO INSTRUCTIONAL PROBLEMS

G.H. ROID
CENTRE FOR LEARNING AND DEVELOPMENT
MCGILL UNIVERSITY

DESIGNING PROGRAMMING LANGUAGES FOR COMPUTER BASED INSTRUCTIONAL SYSTEMS IN RELATION TO INSTRUCTIONAL PROBLEMS

SESSION 9A
ROOM/SALLE 145

By/Par: G. H. Roid,
McGill University,
Université McGill.

Many of us in instructional technology all too often have "solutions" and search for instructional problems to fit them. If the reverse were true and educators brought their problem to us before computer-based systems were designed a number of new design principles would be needed. This problem is particularly important in the design of author languages because they must span the gap between the instructional problems of non-computer people and machine implementation.

Language design has, in the past, been guided more by programming considerations than by the need to implement a wide variety of instructional solutions. Particularly when instructional problems and their solutions dictate the need for simulation or sophistication multimedia display, most of our present author languages are inadequate.

The author of this paper is currently involved in experimentation and development of new author language capabilities with these problems in mind. Experience consulting with teachers and professors about instructional problems has influenced these developments. This work is described and discussed.

LA CONCEPTION DE LANGAGES DE PROGRAMMATION POUR ORDINATEURS AU SERVICE DE L'ENSEIGNEMENT EN FONCTION DES PROBLÈMES PÉDAGOGIQUES

Nous sommes nombreux, en technologie pédagogique, à avoir des "solutions" et à chercher des problèmes pédagogiques qui s'y adaptent. Si l'inverse était vrai, c'est-à-dire que les enseignants nous amènent leurs problèmes avant l'élaboration des systèmes à base d'ordinateurs, il nous faudrait alors certains principes nouveaux de conception. Ce problème est d'une importance particulière lorsqu'on en arrive aux langages utilisés par les pédagogues car ces langages doivent constituer la relation entre les problèmes pédagogiques des enseignants et les conditions imposées par les machines.

Anciennement, la conception des langages a été orientée plus par les besoins de la programmation que par le besoin d'établir une grande diversité de solutions pédagogiques. C'est surtout quand les problèmes pédagogiques et leurs solutions demandent une simulation ou des affichages multiples et compliqués que la plupart de nos langages actuels sont inappropriés.

L'auteur de cette communication s'occupe actuellement de l'expérimentation et du développement du potentiel de nouveaux langages. De consulter les enseignants sur ces problèmes a influencé les développements et c'est ce que nous décrivons et que nous discutons.

Designing Programming Languages for Computer Based Instructional Systems in Relation to Instructional Problems

G.H. ROID

McGill University

Consulting on problems in instructional technology involves an interesting bit of self-control. One of the most difficult things about helping people with their instructional problems is that solutions come too quickly and too easily. Many of us all too often have developed "solutions"and then search for people with problems to fit them. Whether we have a computer, media equipment or programmed instruction in hand, it is difficult to resist advertising it before we have adequately defined a "customers" problem.

If an educator with a problem like "students with different levels of preparation for our program" comes to a CAI consultant he is likely to get an individualized CAI system. If he goes to a media resources centre, he may get study carrels equiped with television,slides, and audio-tapes. If he goes to a group-process expert he is likely to get sensitivity training groups. I guess we all have learned to "do our own thing" with unfailing energy. This problem is particularly important in the design of author languages because they must span the gap between the instructional problems of non-computer people and machine implementation. Author language design has, in the past, been guided more by the need to program a small range of techniques rather than the need to implement a wide variety of instructional solutions. Particularly when instructional problems and their solutions dictate the need for simulation or sophisticated multimedia display, most of our present author languages leave us unprepared.

The present paper first discusses what constitutes a good statement of an instructional problem. Then, applications of computers and author languages are discussed in relation to several examples of instructional problems.

Instructional Problems

Instructional problems are descrepancies that occur when someone who needs to be taught something has not yet been taught effectively. If a teacher has an image of what a student should be able to do upon completing a lesson or program, and the student does not fit that image, a descrepancy exists for the teacher. If a learner wants to gain a new skill but has not yet found a program that deals with it, a problem exists for the learner.

Instructional problems are not always clearly identified or described. It is imperative that they are clearly stated, however, before a solution, particularly a technological one can be tailored for them. Problem statements made by educators sometimes lack observable evidence, data or descriptions of student activities. For example, one might hear the complaint, "my students don't understand geography because we have so few detailed maps to show them". It is not clear what is meant by "understand geography", in terms of what students do or do not do. Perhaps lack of discussion in class, poor test scores or other evidence is present, but is not stated. A prematurely diagnosed cause is included -- "we have so few maps". One of the first tasks of the instructional consultant is to help the teacher or learner define his problem with clarity and observable evidence. When this has been done a difficulty may still remain. It may be that the identified descrepancy in learning is not really an instructional problem. A distinction can be made between a knowledge deficit and a performance deficit (Gilbert, 1967, Geis, 1970). Instruction may be necessary only if a knowledge deficit is present. If a performance deficit is present, the learner has a skill or a set of understood concepts, but does not use them in the environment in which he lives or works. Something as simple as a checklist of steps in a work task or changes in incentives for action can bring out unused skills or knowledge. A common example of a performance deficit in some cases is lack of discussion in a classroom. It may be that students know enough concepts (and are skilled at talking) but for a variety of reasons incentives to discuss are absent (or punishment for saying the "wrong thing" is present).

Programming Languages and Instructional Problems

If consultants and educators were to start with well formed statements of instructional problems, we would find a need for a wide variety of computer programming languages and systems. We would also find that our hardware or currently available languages cannot provide a complete solution in many cases. Several examples from the author's experience in consulting with university instructors may provide clarity here.

The instructor in a biochemistry course in Medicine came to the author to find out more about a computer-based system for self-testing. He had seen the system in operation at another university and wanted to know if it was feasible to consider adopting it locally. A series of discussions were held to identify instructional problems. The instructor indicated that his major concern was that students delayed studying until a short time before exams and did poorly. This was true even though copies of previous exams were posted long before test time. He felt that a self-testing system which actually required the student to give a response before seeing the answer to a question might be a solution.

One could imagine a computer-terminal system which would present questions from a question bank and give feedback to the student as he responds. Any of the frame-by-frame author languages would be suitable for programming such a system. But wait! Is not the question answering itself only a part of the instructional problem? It seems that no matter what system was devised for presenting questions the student has to go to it regularly in sufficient time before the exam for the problem of "cramming" to be solved. This is a problem in self-management. Either the students lack skills in managing their time or there are not adequate incentives to test themselves early and regularly. So, another element in a solution to this problem may be non-computer related. It may be some type of training in progress plotting, or some use of deadlines and incentives (e.g., a portion of the course grade) for self-testing.

In another example, the author worked as an outside experimenter for a large psychology course. An instructional problem was defined by the low mid-term examination scores achieved by some of the students. These students were apparently not responding well to the lectures, conferences and readings in the course -- but due to some as yet unidentified problem. Since an experiment in the use of time-shared computer terminals to present CAI lessons (of a programmed-instruction type) in the course was planned anyway, the underachieving students were chosen as likely subjects. An author language similar to LYRIC was programmed for the time-sharing system at McGill University (Roid, 1971). The language compiler was written in FORTRAN to speed development (but not, of course, execution time!) A study of the students, their learning, and their evaluation of the CAI lessons used, revealed that the lessons were effective because they simplified the language used in the course's textbook and provided many common examples of concepts. This appeared to be due in part to the fact that most of the students in the experimental group did not have English as their first language. Thus, the true instructional problem seemed to be more specifically related to a descrepancy between the reading ability of the students and the level of the text. The use of the computer and any author language, *per se*, would not insure that this problem was solved. Perhaps an author language which provided functions needed to help the educator assess the reading level and rates of students would help in such a situation. However, even if a language did, it would have to allow lessons to be "re-written" to conform to the proper level -- a job of such complexity that might best be done off-line.

The author language developed for this experiment was called MULE (McGill University Language for Education). It was used in an augmented form in other settings. In one such case Education students were introduced to CAI by using MULE to code questions for weekly quizes in their course. Part of the instructional problem was the students lack of familiarity or skills in the use of the computer in education. However, another less specific problem was reflected in the need for weekly quizes. The compiler and executive program for the MULE language had to be altered to allow for a random sample of quiz questions (coded in the MULE language) to be drawn from a weekly pool of questions. The relative simplicity of the author language proved to be an advantage in this setting. Students were able to gain skills in using CAI quickly and easily.

In another situation, computer aids were heavily used in the author's course on multivariate data analysis in Psychology. Two of the instructional problems identified in this course were:

1. Students had no skills in using computers, or computer terminals.
2. Students had no skills in analyzing data using computer programs.

That the first problem could not be solved by CAI or an author language should be obvious. However, once a minimal level of skill in operating a computer terminal and running specifically named programs was attained, a set of CAI lessons in computer system commands and usage were helpful as reference material.

For the second problem, a set of interactive statistical programs were found already available on the university time-sharing system. These were not completely "instructive" in that some terminology and statistical knowledge was assumed for all of them. The development of completely adequate lessons with statistical computations build in would have required a high level author language. APL which is only now available at McGill University might have been appropriate. However, this particular installation of APL does not allow for large sets of data to be stored easily and operated upon by an APL program in the fashion needed in some multivariate problems. The packaged programs already available ran on a system (McGill RAX) which has limited, but quite excellent data file capability. In one case, new programs had to be written in FORTRAN (e.g., a program which generated large Monte Carlo data sets for input to canonical correlation analyses).

One final example again illustrates the need for flexibility in programming languages. The author has worked with a Canadian firm designing graphic terminals for CAI and other applications. Software is needed to demonstrate the capabilities of these terminals and to implement lessons on them. These terminals will undoubtedly be used in a variety of educational settings in which they will be connected to a variety of computers. A tentative solution to this need is to develop modules (subroutines) in a problem-oriented language (e.g., FORTRAN) which do elementary operations such as plot lines or dots on the CRT screen. Although such functions exist in other languages now (e.g., TUTOR), they can not be directly implemented now on this new terminal hardware. Lessons could be constructed using the modules developed, either through a specific program (i.e., one written in FORTRAN) which calls up the modules, or through the use of a compiler or interpreter (written in FORTRAN) which implements an author language which indirectly references these modules.

The Future of Author Languages

It seems that a CAI center, school or college would be advised, for the present to maintain flexibility in the kind of programming languages available to aid instruction. This has been suggested by Zinn (1971). Heavy committment to one author language (of the type now available) would appear to unduly restrict the type of instructional problems that could be attacked. Furthermore, it seems that general-purpose languages such as those found on time-sharing systems (FORTRAN, BASIC, APL, and many others) may still offer considerable power to solve a variety of problems, eventhough they are less convenient to use than author languages (e.g., COURSEWRITER, LYRIC, PLANIT, TUTOR and many others). Perhaps we as educators can be influencial in encouraging the development of more general-purpose languages (and specific compilers) for time-sharing systems. For example, to the authors knowledge, there is currently a lack of implementation of interactive languages such as JOSS, or a string-processing language like SNOBOL, or a list-processing language or a simulation language on time-sharing systems (see Zinn, 1971). Nor is there wide spread availability of hybreds of these languages which might include functions commonly used in CAI programs.

In summary, just as a simple carpenter must maintain a chest full of tools for many purposes, educational technologists of the future will have to maintain a multiplicity of programming languages designed for a multiplicity of instructional problems.

References

Geis, G.L. Premature instruction. *Educational Technology*, 1970, Vol. 10, No. 14, 24-30.

Gilbert, T.F. Praxeonomy: A systematic approach to identifying training needs. *Management of Personnel Quarterly*, 1967, 6, 20-30.

Roid, G.H. User's guide to "MULE" (McGill University Language for Education): A CAI author language. Centre for Learning and Development, McGill University, 1971.

Zinn, K. Requirements for programming languages in computer-based instructional systems. Organization for Economic Cooperation and Development, Centre for Educational Research and Innovation, Paris, March, 1971.

THE USE OF APL
IN COMPUTER ASSISTED LEARNING

by P. P. M. Meincke

ERINDALE CAMPUS, UNIVERSITY OF TORONTO
Mississauga, Ontario

THE USE OF APL IN COMPUTER ASSISTED LEARNING

SESSION 9B
ROOM/SALLE 145

By/Par: P.P.M. Meincke,
Erindale College,
University of Toronto.

A computer assisted learning system using APL has been developed at Erindale College of the University of Toronto. The use of APL has a number of distinct advantages including cost, computational capability, compatability with existing and future systems, and extensive flexibility. A number of institutions, including Orange County Community College District, have had considerable success with APL for CAL. In this paper the author-language presently in use will be described along with some of the programmes which illustrate the powerful advantages of APL. A number of objections to the use of APL have been raised, and these include workspace size limitations, lack of records, and system error messages. These points will be discussed in the light of experience at Erindale which has shown that these objections can be easily overcome.

It is of course unlikely that APL will be the ultimate language for CAL but there are certain characteristics such as the ability to define specialized functions within the workspace which have important implications for future developments, and these will be discussed. Moreover, the programmes are stored as literals which can be easily translated into other systems so the work put in to the pedagogy of a programme is not lost. At Erindale, translators and simulators are being written in APL which will permit the use of many of the programmes already available in such languages as CAN, developed by the Ontario Institute for Studies in Education.

Although the system does not yet have the full capability of the IBM 1500 system in terms of audio tapes etc., APL is being used to drive a plotter and a 48,000 item random access microfiche. There is no reason in principle why other facilities cannot be driven by APL if and when the need arises.

The experience to date has only been extremely positive. The possibilities which are made readily accessible by APL are very exciting, and we are convinced that we have only begun to scratch the surface in exploring the potential of APL in computer assisted learning.

L'UTILISATION DE L'APL EN EAO

On a mis au point un système d'utilisation des ordinateurs dans l'enseignement, basé sur l'APL, au Collège Erindale de l'Université de Toronto. L'APL présente des avantages particuliers dont le coût, les possibilités de calcul, celles d'adaptation au système actuel et à venir et sa très grande souplesse. Certains établissements, dont le Orange County Community College District, ont très bien réussi en se servant de l'APL pour l'EAO. Dans cette communication, nous décrivons le langage de programmation des cours et certains programmes illustrant les grands avantages de l'APL. On a élevé des objections au sujet de l'APL comme, par exemple, ses limites de mémoire, son manque d'informations enregistrées et ses messages d'erreurs. Nous en parlerons en fonction de notre expérience qui, à Erindale, a montré que ces objections peuvent être aisément réfutées.

Naturellement, il est peu probable que l'APL deviendra le langage ultime des ordinateurs pédagogiques mais ce langage présente certaines caractéristiques qui sont importantes pour l'avenir, dont nous discuterons et dont nous donnons, comme exemple, la possibilité de définir des fonctions spécialisées dans le domaine du travail. De plus, les programmes sont mis en mémoire sous une forme qui peut être facilement transposeé dans d'autres systèmes de sorte que le travail pédagogique d'un programme ne sera pas perdu. À Erindale, les travaux de traduction et de simulation sont formulés en APL ce qui permettra de se servir de nombreux programmes existants dans des langages tels que le CAN mis au point par l'Institut ontarien de recherche en éducation.

Quoique le système n'ait pas encore la puissance du système IBM 1500 en matière de bandes phoniques, etc., l'APL sert à commander une traceuse et à consulter des microfiches au nombre de 48000 et aléatoirement. Il n'y a aucune raison, en principe, pour que l'APL ne commande pas éventuellement d'autres installations.

À ce jour, notre expérience n'a été qu'extrêmement positive. Les possibilités ouvertes sans difficulté par l'APL sont très intéressantes et nous sommes persuadés que nous ne venons que d'explorer superficiellement les possibilités de l'APL dans l'enseignement à l'aide d'ordinateurs.

THE USE OF APL IN COMPUTER ASSISTED LEARNING

by

P.P.M. MEINCKE

INTRODUCTION

When the spirit moves one to an interest in Computer Assisted Learning (variously known as Computer Aided Instruction, Computer Assisted Teaching etc.) one is faced with a bewildering galaxy of systems. There are COURSEWRITERS, CANS, TUTORS, CALAPLS and so on. After a little investigation, it soon becomes apparent that there are two major ways of becoming involved : If the individual is good at finding outside support, he has, up to now, been able to find a few hundred thousand dollars or even a few million to set up a major installation. IBM has supported a number of COURSEWRITER installations , but some of these major installations have found it difficult to continue after outside support is withdrawn.

The other way of becoming involved is to buy or rent a few terminals and set up a communications link to a major installation. This method is the one which the National Research Council has been advocating, and they have set up an extensive facility in Ottawa into which any interested parties can connect their terminals. Indeed, the directors of most major installations have ,up to now, encouraged others to use their facilities as long as there was capacity to spare. But there are some minor difficulties with such arrangements. The first problem is of course getting an invitation to join the system, but this seems to present a minor difficulty to most people. There are problems such as the lack of control over the system, and one is essentially at the mercy of the major interests in the system. Since changes would be difficult to make in most systems to which a large number of users are connected, the ability to innovate can sometimes be severely curtailed. This is not true of all systems but certainly should be a consideration in making a decision to proceed in this manner. The main advantage of this kind of operation is that it allows one to keep up to date with what is going on in the major system.

There is another possible way to explore CAL which we have found most useful at Erindale. Basically , the idea is to use a powerful, high level interactive language on which to build the CAL system. At first sight, this may seem to be a very expensive way to run a CAL system, but the actual running costs, although they are certainly greater than a system specifically designed for CAL, are not that far out of line and there is a great saving in development costs. In addition, one is using the same terminal for a number of things and this also helps with the cost factor.

There are at least two essential ingredients, however. One is a highly cooperative and efficient Computing Centre which keeps the Interactive System going with a high degree of reliability, and we have been very fortunate in having the support of the University of Toronto Computing Centre. Another essential ingredient is at least one bright creative expert in the interactive language. In this area, we have been very fortunate in having Mr. David Vaskevitch, who has played a central role in setting up this system while enrolled as an undergraduate at the University of Toronto. He is giving a paper in this Conference on the work he is doing in setting up programs to teach Symbolic Logic.

When I originally decided to use APL as a basis for CAL, the decision was based on the potential flexibility of such a system, the relatively low cost, the fact that we could operate at any level from one terminal to more than we could afford to rent at that time, and the fact that APL was a highly reliable interactive system which required no effort on our part to maintain. It seemed to me that APL was much like a piece of commercially available, reliable but highly versatile research equipment which could be used to explore many facets of this rapidly developing field quickly and easily even though it might not do any one job as quickly and as efficiently as a specially designed tool.

Since then, David has developed the original package extensively and we have only recently started to make direct comparisons with other systems. We have found to our surprise that this system based on APL compares very favourably with other systems in such respects as cost, number of students, response time, and others which were not given as much weight as flexibility and reliability in the original decision. For example, if we analyze the cost in the same way many others do, we find that a system based on APL used by 100 students at a time could cost just a little over $1 per hour. Moreover, we have direct empirical evidence that an APL system can be used by more than 100 people simultaneously without objectionable delay in response time. It is not possible to go into the details of an APL system at this time to show how this is possible, but a few of the more general features of the system will be described.

SOME CHARACTERISTICS OF APL

APL, (A Programming Language) is a very high level programming language with a high degree of internal logic which is especially suited to interactive programming. It is an interpretive system and as such can be rather inefficient in its use of computer time, but it is estimated that APL reduces the time required for program design by at least a factor of five. It is an extremely forgiving language and allows even the most inept programmer to get something going even if it is inefficient.

Now, all this talk of inefficient use of the computer makes most computer experts wince visibly, but I am convinced that we are very rapidly reaching a trade-off point between human effort and computer effort. Certainly, it is true that one cannot afford to be inefficient in the huge and repetitive number crunching operations, but there is a very great need for "one shot" computer programs and APL fills this need admirably. This interactive system is particularly useful in designing program systems as has been shown in business applications, but even there it has been found economical to use APL after the program design is complete.

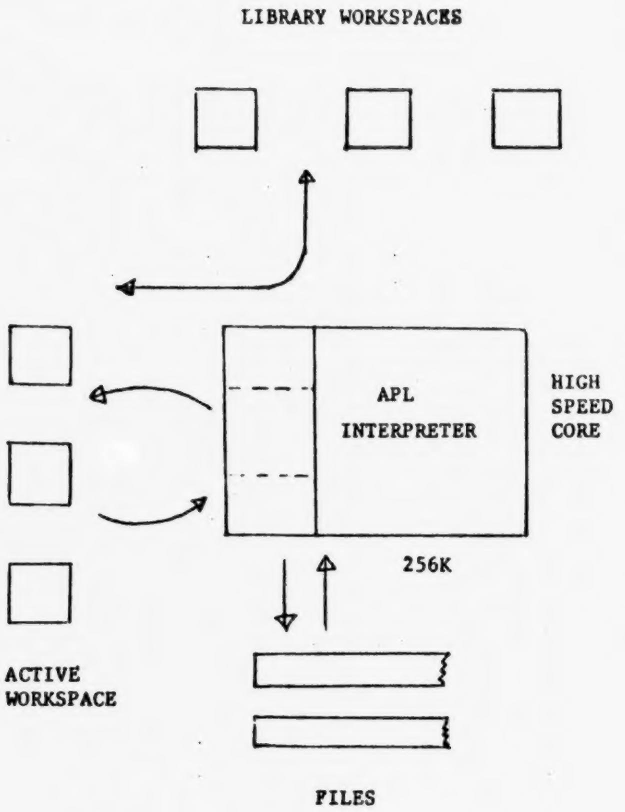
There is still a great deal to learn about computer assisted learning and it is my contention that the system which is used to explore this rapidly developing field should place as few barriers as possible in the paths of those who wish to explore uncharted paths. The system should allow this initial exploration to take place as easily as possible. If new and fertile land is discovered, the inefficiencies of the programming can always be tidied up later to conserve computer effort in the event that large numbers of students are using the program. Even this rather cavalier attitude toward inefficient use of the computer has not really given rise to any real evidence of waste. There is far more wasted effort arising from inability to write decent lesson material.

APL is an interpretive language, that is it interprets each line as it comes along, and executes the operations contained in that line. APL does not compile a program as do most systems, and therein lies a certain amount of its inefficiency. However, this interpretive characteristic does lend itself to interactive programming. Another saving grace is that the programming can be very sophisticated and each line can contain a very large number of operations so the number of lines in a program is drastically reduced which also reduces the running time. In fact most APL programmers spend a good deal of their time comparing how much they can pack into one line of APL. The standard phrase is "Oh, I did that in one line."

The system is outlined in figure 1. A rather large amount of high speed core, (about 250,000 bytes), is required although York University has developed a version which will run on only 76,000 bytes. When a user signs on, he is assigned an active workspace of 36,000 bytes which is stored on disc until some action by the high speed core is required. At that time, the active workspace is put into a slot in the high speed core where the APL interpreter then acts on the instructions stored in the workspace.

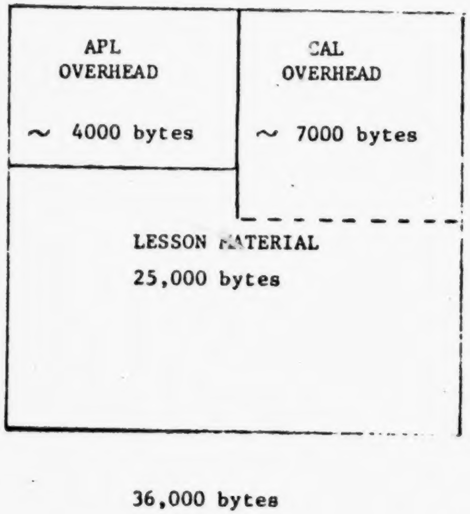
There are libraries of workspaces also stored on the discs and therefore available at all times. The user does not have to wait for tapes to be loaded or discs to be brought up. He simply loads a copy of the workspace stored in the library into his active workspace and that copy is uniquely his until he signs off or stores the workspace in his own library.

If many students are using the same lesson at the same time, there are many duplicates of the same lesson, one for each student. This duplication is wasteful if the CAL periods are to be highly scheduled ,and , indeed, most systems specifically designed for CAL store lesson material in a space which is common to many students. It is our feeling that the freedom gained for the learner is worth the extra cost in the operation of the computer. The learner is free to roam through the entire library and does not have to go to the terminal only when his lesson is available during the scheduled hour. Moreover, many of these more tightly designed systems start to run into limitations when more complex manipulations are encountered.


LIBRARY WORKSPACES
APL
INTERPRETER
HIGH
SPEED
CORE
256K
ACTIVE
WORKSPACE
FILES

F I G U R E 1

THE APL SYSTEM


APL
OVERHEAD
~ 4000 bytes
CAL
OVERHEAD
~ 7000 bytes
LESSON MATERIAL
25,000 bytes
36,000 bytes

F I G U R E 2

AN APL WORKSPACE

The workspace for a CAL lesson is divided up as shown in figure 2. Approximately 4000 bytes are required in every workspace for APL overhead. The latest version of our CAL requires about 7000 bytes to store the functions for the operating codes. This 7000 bytes can be altered by the individual author who may not wish to use all of the operations codes available or may wish to write some of his own. Again, it may seem wasteful to store duplicates of the operating code functions in each workspace, but this means that each workspace is a completely self contained and independent entity. It can be changed without affecting the rest of the system in any way or it can be transported to another APL system and work just as well as it did on the original system. This tremendous flexibility is just what is needed to stimulate the innovation required in this field.

Another problem sometimes arises when the lesson material will not all fit into one workspace. Normally, the material in one workspace will occupy a student for about one hour, but if there is a great deal of remedial material, the workspace size can be a limitation. The lesson can always be split into two or more workspaces but this normally requires the student to load the next workspace which can be a nuisance and somewhat distracting. The UNQUOTE operator developed at York University helps out enormously in this situation because it makes it possible to write a command in one workspace to load in another , so one effectively has no limitation on the amount of material in the lesson. This facility will be available in other APL sytems soon. In fact this facility makes it possible to develop a PROCTOR which will converse with the student, try to find out what the student would like to learn while letting him know what is available on certain subjects and then automatically load in the appropriate material. Such a system has been developed by David Vaskevitch for Ryerson Polytechnic Institute which uses York APL.

The APL system also has file capability which will allow the gathering of data on various lessons and on student performance if necessary. A mailbox system has also been developed which will allow the student to send through the computer terminal any comments he has about the lesson directly to the author. Files could also be used to store common operating routines if space should become a problem.

The number of users that can be adequately handled by the system is determined by the computer and disc drives. We do not yet have enough experience with the University of Toronto system to determine the maximum number of users, but our experience with a commercial system indicates that more than 100 users can be serviced without objectionable response time. Apparently, storage space is not a problem since disc packs can be added on to the system as long as the money holds out. (Actually the limit is 2 to the 48th power.)

APL uses symbols rather than English language expressions for its operators and although this makes it much more universal, there is the problem of learning the meaning of the symbols. The mathematical expressions are concise, and it is possible to perform very complex operations with very simple looking expressions. However there is a very real need for a simple operations code for writing lessons. These codes can be written as functions in APL and naturally , the name of the function can be in any language and it will continue to operate on the "language independent" APL base.

THE COURSE AUTHORING LANGUAGE AND OPERATION CODES

It is recognized that many instructors who may wish to write lessons will not necessarily be skilled computer programmers and the CAL systems usually have a simple course authoring language which does not require skill in computer programming. Some, such as COURSEWRITER and A TUTORIAL SYSTEM, (developed at Cornell University with support from IBM), have the answer processing built right in to the system and it is only necessary for the author to type in the questions, right answers, and wrong answers with appropriate replies. So far, we have found that most of the authors want to have more control over the actual structure of the answer processing and structure of the lesson even though it requires that they exercise care in the writing of branches and providing labels on statements but we certainly intend to write a conversational package for writing simple programs when the need arises.

The real problem in course authoring languages seems to be that the author soon runs into the limitations imposed by the system and the language. He wants to do something just a little different, but cannot without changing the system. With APL, this kind of change is much easier to make. The author can usually learn enough APL to write his own functions for his particular need or else they can be written by someone else in a very short time. These functions may not be the most efficient, but they will certainly serve the purpose and provide a very adequate basis for testing the ideas involved. Moreover, they can be entirely contained in a particular workspace without disrupting the rest of the system apart from the inconvenience experienced by some users because the format is not identical to the other lessons. In most other systems , it is necessary to have an assembler level programmer make the changes involved and it is usually necessary to obtain the consent of the other users of the system. We have found that innovation is definitely stimulated by this freedom from restrictions.

The operations codes in the latest version of NEWAPLCAT are shown in figure 3. Most of them are reasonably self explanatory to those who have done any work with computer assisted instruction ,but some of them will require more detailed explanation. Literal output to the student is simply put in quotation marks as in basic APL. If the answer is to be interpreted as literal data, then the READ statement is used while numeric data is accepted with the READN statement. These statements also automatically look for student control commands such as END. The power of APL is demonstrated in the READN statement which will accept not only a number but any valid mathematical expression. (There are problems with the student who is supposed to be drilling in multiplication when he discovers that all he has to type in is 3x4 instead of 12.)

The MATCH command is used to check if the student's response is identical to the answer required, but care must be exercised because even blanks will be counted unless the IGNORE instruction is used. This instruction is generally used only for numerical responses since it will accept any valid mathematical expression. A range of error can also be included.

The SCAN instruction is used much more frequently in literal data analysis and it will look for the occurrence of a particular set of characters in the response. Alternative sets can be specified with the OR command and other required sets can be specified by the AND instruction. One very powerful addition is the BEFORE instruction which specifies the order in which sets of

READ READN	Produces a ⍞ (⎕ for numerical data) on the student's terminal to indicate a response is required. Also looks for student command words such as END. Stores the input from the student in systems buffer BO. READN will accept any valid mathematical expression.
IF	The heart of the conditional branch statement → L1 IF SCAN 'RIGHT ANSWER'
SCAN 'X'	Looks for the occurrence of the character string X in the student's answer.
MATCH 'X'	Looks for an exact match (including blanks) of X or the numerical value of N.
'X' AND 'Y'	Requires the occurrence of both X and Y.
'X' OR 'Y'	Looks for either X or Y.
'X' BEFORE 'Y'	X must appear before Y.
IGNORE REPLACES	To massage student answer, e.g. IGNORE A removes all A's from student answer.
COMP	Frees the terminal to be used for APL but keeps program available for student.
DATE	Prints out date.
→ START	Initializes program.
STOP	Final statement.
→ LASTREAD	Will branch to the last executed read statement.

Figure 3

SOME OPERATION CODES

characters must appear. The computer statement to look for the physical unit "FOOT POUND" would then be: SCAN 'FOOT' OR 'FEET' OR 'FT' BEFORE 'POUND' OR 'LB' This simple statement now looks for and accepts at least twelve possible answers including such things as FT-LBS.

The situation can be improved still further by editing the student's responses with the IGNORE and REPLACE instructions. For example the IGNORE instruction could be used to eliminate all occurrences of e or o in a student's response and therefore accept FT or FOOT or FEET. In this respect, this system based on APL certainly compares very favourably with other course authoring languages such as Coursewriter and the CAN4 language developed by OISE.

The set of operating codes has developed through several stages over the past year, but all the lessons written with even the earliest codes are still in active use. The main differences between most languages for CAL lie not in the answer processing capabilities but rather in the convenience with which rather complex comparisons can be made, and the ease with which lessons can be changed and edited ,and in this respect, the present set of codes is as sophisticated and convenient as any presently available.

It is usually informative to compare how a simple lesson would be written in various systems. Figure 4 illustrates how a simple drill in multiplication would be written in our system.
Even this simple example shows the advantage of having a powerful interactive computing language as the base on which to build the CAL system. All the complex mathematical operations are readily available and formatting and space reservation are all handled by the APL system. The advantage becomes a necessity if more complicated mathematical objects such as vectors or matrices are involved.

A list of the progams which are presently available on the system is given in figure 5.

EQUIPMENT

Unfortunately, the range of terminals which can be used with APL is somewhat restricted at present because the transmission code necessary for the full character set is not compatible with many of the available terminals. The one in use at Erindale at present is the rather bulky and

```
      ∇MULTIPLY [⎕]∇
    ∇ MULTIPLY
[1]    →START
[2]    'HELLO ',B1,' LETS PRACTISE
       MULTIPLYING NUMBERS LESS THAN 13'
[3]   Q:A←?12
[4]    B←?12
[5]    'WHAT IS ';A;' TIMES ';B;'?'
[6]    READN
[7]    →R2 IF MATCH A×B
[8]    →A1 IF MATCH A+B
[9]    'SORRY ',B1,' TRY AGAIN.'
[10]   READN
[11]   →R2 IF MATCH A×B
[12]   'THATS STILL NOT RIGHT. THE ANSWER
       IS ';A×B;' TRY THIS ONE.'
[13]   →Q
[14]  A1:'YOU HAVE ADDED. TRY AGAIN.'
[15]   →LASTREAD
[16]  R2:'THATS RIGHT ',B1,' NOW'
[17]   C←?12
[18]   'WHAT TIMES ';B;'IS EQUAL TO ';B×C;'?'
[19]   READN
[20]   →R3 IF MATCH C
[21]   'SORRY, THATS NOT RIGHT, TRY AGAIN.'
[22]   READN
[23]   →R3 IF MATCH C
[24]   'THAT IS STILL NOT CORRECT. THE
       RIGHT ANSWER IS ';C;' TRY THIS.'
[25]   →Q
[26]  R3:'RIGHT ',B1,' NOW,'
[27]   →Q
[28]   STOP 'MULTIPLY'
    ∇
```

FIGURE 4

A SIMPLE DRILL

WS NAME	LESSON NAME	DESCRIPTION
STUDENTHOW	STUDENTHOW	HOW TO USE THE TERMINAL FOR CAL
WFFNPROOF	WFFNPROOF	AN INTRODUCTION TO THE POWER OF SYMBOLIC LOGIC - THE TARDY BUSDRIVER
SHAKEAWFF	SHAKEAWFF	DEFINES WFFS (WELL FORMED FORMULAE) PRACTICE IN WFF RECOGNTION
GAMES	GAME1	WFF RECOGNITION
GAMES	GAME2	PLAY SHAKEAWFF WITH THE COMPUTER. IT WILL EXPLAIN YOUR MISTAKES
MOTION	MOTION	BASIC IDEAS OF MOTION
CIRCULAR	CIRCULAR	INTRODUCTION TO THE PHYSICS OF CIRCULAR MOTION
CIRCULAR2	CIRCULAR2	CONTINUES A STUDY OF CIRCULAR MOTION
WORKA	WORKA	INTRODUCTION TO AND PRACTICE IN THE CONCEPTS OF WORK AND ENERGY
APLCAT1	WORK	A SKETCHY LOOK AT SOME ASPECTS OF WORK
MICHELSON	MICHELSON	BASIC IDEAS OF THE MICHELSON MORLEY EXPERIMENT (RELATIVITY)
MORLEY	MORLEY	BASIC IDEAS OF THE MICHELSON MORLEY EXPERIMENT (PART 2)
LORENTZ	LORENTZ	THE LORENTZ FITZGERALD CONTRACTION
RELATIVEXPT	RELATIVEXPT	TO ACCELERATE VARIOUS PARTICLES TO RELATIVISTIC SPEEDS TO SEE MASS VARY
APLCAT1	LOADLINE	LOADLINE ANALYSIS FOR FET AMPLIFIERS. TEXT BY BROPHY REQUIRED
APLCAT1	LOADLINETRANS	ANALYSIS OF TRANSFORMER COUPLED FET AMPLIFIER
APLCAT1	MULTIPLY	DRILL IN MULTIPLYING NUMBERS LESS THAN 13
MATHDRILL	MATHDRILL	DRILL IN SIMPLE ARITHMETIC
LIMITS	LIMITS	INTRODUCTION TO THE NOTION OF LIMITS
CALCULUS1	CALCULUS1	BASIC CALCULUS PART ONE
CALCULUS2	CALCULUS2	BASIC CALCULUS PART TWO
CALCULUS3	CALCULUS3	BASIC CALCULUS PART THREE
EANDM	EANDM	BASIC CONCEPTS OF ELECTRICITY AND MAGNETISM
UNITS	UNITS	DRILL IN THE BASIC UNITS IN PHYSICS
APLCATHOW1	APLCATHOW1	WRITING LESSONS IN APLCAT, PART ONE
APLCATTWO	APLCATTWO	WRITING LESSONS IN APLCAT, PART TWO
APLTHREE	APLTHREE	WRITING LESSONS IN APLCAT, PART THREE
NEWAPLCAT	DESCRIBE	DESCRIBES THE OPERATING CODES IN THE LATEST VERSION OF APLCAT

FIGURE 5 A LIST OF PROGRAMMES AVAILABLE FOR CAL

noisy 2741 communications terminal. It is hoped that the manufacturers will soon produce equipment which can handle the APL character set. In the meantime, software support for teletype ports can be implemented. Although this is not entirely satisfactory for those who want to use the full range of APL characters, it is a very acceptable solution for the Computer Assisted Learning program which seldom requires more than the normal character set. We plan to experiment with a number of the very small portable terminals which can be taken home and hooked up to the television set and the telephone so that the student can actually do some of this material at home provided he can negotiate for the use of the television and the telephone for the required period of time.

Many of the systems which were designed specifically for CAL have facilities for light pens, audio playback,slide projectors etc. APL can be used to drive such devices as well when the need arises. XY plotters can easily be interfaced with the system, and in the Orange Coast College District they are successfully running random access microfiche devices capable of accessing any one of 48,000 slides in less than four seconds. Such devices will undoubtedly cut down on the cost of storing highly static literal information and make the use of pictorial information very feasible.

There are a few difficulties with interfacing APL with graphics terminals at the present time, but there is a very active program at the University of California at Irvine to explore these possibilities. One of the difficulties we found in one of the CRT terminals was that attempts to underline letters simply erased the letter and left the underlining. Moreover, graphs slowly disappeared off the top of the screen as the lesson progressed line by line, so it was not just a simple matter of taking over the lessons which had been written for the hard copy typewriter terminals into the CRT format. There is no doubt that students do like to take the hard copy from a lesson away with them, so it may not be too wise to press too hard for the display-only terminal .

The ultimate terminal will of course contain a great deal of local buffer storage so that special character sets can be transmitted down the line,stored locally and accessed with 8 bit codes or less. This feature or something close to it is already available in the PLATO IV terminal. Ultimately, it should be possible to transmit and store locally the entire information necessary to display colour pictures. A hard copy facility right at the terminal should also be available. Such terminals are undoubtedly a few years off but not that far away. Certainly the technology is available today but we still have a lot to learn about how best to use it and how we should be packaging material even for the terminals we have available today. How does one take into account for example, the vast range of responses on the part of the students when the computer calls them by name? We think that we can make some progress along these lines even with the rather limited equipment that is available for APL right now, and perhaps we can even learn a little about how the terminal of the future should be designed to make it most effective as an interface between the user and the computer-data bank.

COSTS

The criticism which is most often levelled at computer assisted learning is that it costs too much and that we cannot afford it even if it were effective. It takes a great deal of time and effort to write good lessons and even more stamina to keep them up to date. Moreover, computers are costly things to run and electronic storage of information is more costly than books.

The usual answer to these criticisms is to point out that the cost of electronic information processing is going down by a factor of ten every seven years while people are becoming more expensive. I do not believe that CAL will replace human interaction in the learning process but that it will have a valuable role to play if we can find out how to use it properly. It is therefore essential to have a good idea of its cost and to realize that there will have to be an initial stage of experimentation during which it will certainly not pay for itself. Nevertheless, it was surprising to me to learn just how little our system costs even at this stage and it is very easy to see in the near future, developments in terminals and computers which will bring this cost down dramatically.

It has been estimated that the PLATO IV system at the University of Illinois will cost less than one dollar per student hour including terminals, royalties,computer and storage. At this rate, CAL is quite competitive with conventional teaching costs even at the elementary school level. In our system, the terminal we presently use rents for $105 per month. The telephone costs $15 per month. The present charge levied by the University of Toronto Computing Centre is $3 per hour connect time and $3 per CPU minute which our experience indicates comes to about $4 per hour for CAL . If the computer terminals are used for 150 hours per month, the total charge is less than $5 per student hour. The computing costs are allocation dollars at present and it is felt that the price is somewhat inflated at the present time. In fact , we have a special rate with a commercial firm which amounts to much less than this if the port is used 150 hours per month.

The cost of terminals will soon be reduced by a factor of at least 2 and possibly 5. If we assume that the minimum cost will be about $50 per month including servicing, and that the terminals are used on the average 8 hours a day for 300 days of the year, the cost of the terminal is then 25¢ per student hour. It is difficult to assess the costs of the computer since the one we are using is now an old model, but it is supporting three interactive systems at the present time. A reasonable figure might be about $120,000 per year which comes to 50¢ per student hour if on the average, one hundred terminals are supported. Communication and lesson costs must be added but it certainly appears as if the cost could be below $1 per student hour.

Another feature of this system is that it can be used for a number of other things. For example, we are developing an interactive student information system so that the terminals can be heavily used by the Registrar's Office during registration and then moved out for CAL during the rest of the year. Terminals could also be provided for staff to write lessons, keep records, place orders, etc. If the system can be used to explore a number of ways in which interactive computing can be useful in the university environment, then the cost is spread even further.

In any case , we feel that the cost we are presently bearing is one which we can afford in order to explore CAL. It is interesting to note that this whole project has received no outside support at all and that it has been built up by concentrating on CAL itself and not on the hardware or assembler level programming.

SUMMARY

In conclusion, we have no regrets for having chosen APL as a base on which to build our CAL system. This powerful interactive language has allowed us to do everything we have wanted to do so far and it is interesting to note that our system could do almost anything specified in the taxonomy circulated by the National Research Council of Canada. The cost calculated for high levels of usage appears to be competitive with specially designed systems and the system can provide excellent response time for large numbers of users. Our experience indicates that a thorough study should be made of APL to determine why the system can provide such low cost and response time, benefits which we did not originally anticipate.

CAL LANGUAGE STANDARDS

J. W. Brahan
National Research Council of Canada

CAL LANGUAGE STANDARDS

SESSION 9C
ROOM/SALLE 145

By/Par: J.W. Brahan,
National Research Council of Canada

Instructional applications of the computer have resulted in the development of many specialized high level programming languages, each one aimed at providing effective access to the computer by student, teacher, course author, and/or researcher. At any given installation, the benefits to be gained from such a language are quite evident. However, the variety of such languages which has evolved has resulted in great difficulty in exchanging CAL programme materials among different institutions. Thus much effort is expended on development of software which exists elsewhere, but not in a form suitable for the particular computer installation at the institution wishing to make use of it.

A common programming language for CAL applications would represent a major step towards permitting exchange of CAL software between different computer installations. To be effective, such a language must take into account the variety of approaches used in instructional applications of the computer. It must consider the requirements of users with a variety of skills. It must include provision for the utilisation of a wide range of specialized input-output devices required to permit effective communication between the user and the computer. As with all standards, provision must be made for the CAL language standard to be updated in order to respond to changes in user requirements.

The development and maintenance of standards for programming languages requires the cooperation of the user, the equipment supplier, and the researcher. As a first step towards arriving at definition of user requirements for a CAL programming language, the NRC Associate Committee on Instructional Technology convened a working panel whose members have participated actively in a variety of CAL application, with a variety of programming languages involved. The task assigned to the working panel was the definition of a functional specification of a programming language to meet user requirements. To arrive at a standard from the functional specification requires the production of a detailed specification followed by implementation at a number of centres. This implementation stage must be completed out before one can say a standard truly exists, since a standard has very little meaning to the user if he cannot apply it.

NORMES DES LANGAGES EAO

Des applications pédagogiques de l'ordinateur ont conduit au développement de beaucoup de langages de programmation spécialisés et d'un haut niveau, chacun ayant pour but de permettre à l'étudiant, au professeur, à l'auteur de cours et au chercheur d'avoir accès à l'ordinateur de la manière la plus efficace. Les avantages obtenus de ces langages, à n'importe quelle installation, sont évidents. Cependant, la diversité de ces langages, qui ont évolués, est à l'origine de la grande difficulté d'échange entre établissements se servant de programmes EAO. Ainsi, on dépense beaucoup d'énergie pour développer une programmation déjà existant ailleurs mais dans une forme qui n'est pas appropriée à l'ordinateur de tel ou tel établissement désirant s'en servir.

Un même langage de programmation permettrait de vraiment faire des progrès et des échanges de programmation EAO entre établissements disposant d'ordinateurs et d'installations différents. Pour être efficace, ce langage doit tenir compte des différents angles sous lesquels les applications sont envisagées. Il doit être conçu en fonction des possibilités des étudiants. Il doit comprendre des dispositions nécessaires pour l'utilisation d'une large gamme de dispositifs d'entrée et de sortie spécialisés pour permettre des relations efficaces entre l'usager et l'ordinateur. Il faut prendre des dispositions pour que les normes du langage EAO, comme toutes les normes, soient mises à jour afin de s'adapter aux nouvelles situations.

Le développement et la tenue à jour de normes de programmation nécessitent une coopération entre l'usager, le constructeur des machines et le chercheur. Un premier pas vers la définition des besoins de l'usager a été fait par le Comité associé de technologie pédagogique du CNRC qui a convoqué un groupe de travail ayant participé activement aux applications pédagogiques en se servant de langages différents de programmation. Ce groupe devait définir les paramètres fonctionnels du langage de programmation satisfaisant les besoins des utilisateurs. Pour obtenir des normes en partant de ces définitions, il faut d'abord détailler les éléments du langage à mettre en ouvre dans plusieurs établissements. Cette mise en oeuvre doit être terminée avant que l'on puisse affirmer qu'une norme existe réellement, car la norme a peu de signification pour l'utilisateur s'il ne peut pas l'appliquer.

The lack of adequate standards in any field makes it difficult to select equipment without fear of premature obsolescence, makes it almost impossible to effectively compare alternative choices, results in higher costs to the consumer and, in general, is a major barrier to the effective exploitation of the available technology. Standards permit the transfer of knowledge and equipment from one group to another. They provide for effective training of workers; skills which are based on standards have a wide range of application. Some measure of the importance of standards is indicated by the fact that more than 1200 standards have been approved by the Canadian Standards Association. These cover an extensive range of topics, with each standard representing a general agreement among maker, seller and user groups as to the best current practice with regard to some specific problem.

Standards are particularly important in fast growing technologies such as the computer field, where the capital investment required can be very large and no organization can afford to isolate itself from the general community by adopting non-standard practices. In such fast growing areas, it is important that standards be introduced as soon as requirements can be adequately defined and the solutions to these requirements tested sufficiently. One could say that these standards are introduced in anticipation of widespread application, but in many cases, these applications could not take place without the existence of the standards.

Computer Aided Learning is one field which is very much dependent on standards if it is to result in extensive and effective use being made of the computer. Of particular importance is the need for standards to permit the transfer of computer-based materials from one organization to another without the necessity of an expensive reprogramming operation. Compared to other instructional materials, computer-based materials are expensive to develop and, in general, must reach a wide audience if they are to be economically viable.

One means of improving the "accessibility" of a computer system to users who are not computer specialists is through the use of so-called high level programming languages, whose characteristics are tailored to certain specific applications. The widespread use of languages such as FORTRAN, BASIC, COBOL, etc., attests to the effectiveness of this approach. This same approach has been applied in the field of Computer Aided Learning. The result in this case has been the development of many specialized high level programming languages, each one aimed at providing effective access to the computer by student, teacher, course author and/or researcher. When one considers any given installation, the benefits to be gained from such a language are very much in evidence. However, the variety of such languages which has evolved has resulted in a major barrier to the exchange of CAL programme materials among different interested institutions. As a result, much effort is expended on the development of software which exists elsewhere, but not in a form in which it can be utilized on the computer installation of the organization wishing to make use of it.

Let us consider briefly, the conditions which should exist in any field before the development of a general standard is indicated. Obviously, there must exist a need, either immediate or future for the function which the standard defines. There must also be an indication that there will be broad application of the standard. Finally, there must be an indication of interest in the activity on the part of suppliers of equipment.

Some indication of the extent of activity in Computer Aided Learning is provided by an examination of a catalog of computer based instructional materials, prepared by Helen A. Lekan of the University of Wisconsin(1). It should be noted that the 1970 edition of the publication has been used so that specific data applying to any given programming language should not be considered current. There have been several new CAL programming languages introduced since the catalog was published and some are no longer available. However, the following data which have been derived from the catalog are indicative of the general situation which currently exists.

The publication lists a total of 96 organizations interested and actively engaged (a few of these are indicated as being in the planning stage) in various aspects of Computer Aided Learning. Of the 96, 87 are U.S. organizations; thus the data are primarily derived from activities in the U.S.

An indication of the range of application areas is provided by the catalog's listing of entries by subject matter which includes 55 subject categories. As one might expect, programmes in mathematics lead the list with 216 entries out of a total of 912. However, it is interesting to note that programmes in the health professions (79) are third most plentiful in the catalog, immediately following those dealing with computer operations and programming.

An indication of the number of computer manufacturers and variety of equipment involved is provided by the catalog in a listing of programmes by central processor. This list includes 53 different computers from 15 different suppliers. The distribution is presented graphically in Fig. 1.

These data would indicate a situation which would benefit from the adherence to standards as much as possible. However, when we examine the situation in regard to programming languages, we find that there are 54 different programming languages listed as being used. It should be noted that not all of these are high level languages, a few are assembly languages. Many of these languages have only one or two entries in the catalog. However, the number of languages which have been used to produce a very considerable amount of course material is greater than one would expect to be acceptable if exchange of such materials is considered desirable. A ranking of programming languages by number of programmes reported is illustrated in Fig. 2 while the number of institutions using a given programming language is shown in Fig. 3. It is interesting to note that several of the languages which are in use at only one institution have been used to generate more programmes than some languages which are in use at several institutions. The outstanding example is the TUTOR language which is used only at the University of Illinois PLATO project. One hundred eighteen programmes are listed as being written in TUTOR compared to 43 CAL programmes in FORTRAN which has entries from 17 institutions. Perhaps some indication of the effectiveness of a language for CAL applications can be obtained from the number of programmes per installation of the language.

In Canada, we cannot lay claim to 54 different programming languages being used for CAL applications, but considering the smaller number of institutions involved, the variety of specialized programming languages in use is rather surprising and an effective barrier to the exchange of course materials. A list of some of the languages being used is shown in Fig. 4.

Benefits are to be gained from the exchange of CAL programmes both at the research level and at the application level. At the research and development level, the exchange of programmes permits more effective development effort by eliminating the duplication of work which often results because materials cannot be exchanged. At the production and application level, the exchange of programme materials spreads authoring costs over a much wider "market" than is otherwise possible. More effort can thus be expended on programme development and evaluation, with the result being a higher quality and more economical product. However, if the variety of programming languages being used for CAL applications continues to increase, and an effective means is not found for exchanging computer based materials, these materials will continue to be considerably more expensive than they might otherwise be. Materials which might be economically viable if they were to reach a large audience could be far too costly to consider using for the small audience resulting from the fragmentation caused by the multiplicity of programming languages.

Two approaches can be considered to provide a means for the exchange of computer based course materials. One could be considered an "individual solution" and the other a "group solution" to the problem.

The individual solution is represented by the use of translators. Thus a given installation would use one or possibly two languages for CAL applications and translate programmes written in other languages. It is technically feasible to produce programmes which will translate from one computer language to another automatically, or at least semi-automatically. However, certain problems can arise in attempting to use and maintain such translators. In general, the user must be familiar with both the source and destination languages to make effective use of the programme material. The translator programme must reflect changes introduced in the languages with which it is concerned. From the practical considerations, it is unlikely that any installation could deal with more than a few languages in this way. Finally, the approach has little influence on the number of CAL programming languages either present or future. In spite of these drawbacks, the use of translator programmes does represent a practical means for transporting course materials from one installation to another in that it can be implemented relatively quickly. However, it is suggested that this approach should be considered, at most, an interim solution and a solution should be sought which resolves more completely the problems.

The so-named "group solution" to the problem of exchange of materials is based on the existence of a programming language common to a large number of different installations and common to a number of different machines. To achieve the development and implementation of an effective common programming language for CAL is not a simple undertaking. It requires the continuing cooperation of users at all levels, and of equipment suppliers.

As a first step towards arriving at a definition of requirements for a CAL programming language, the Associate Committee on Instructional Technology convened a working panel with the task of defining the functional specifications of such a language. The members of this panel have participated in a variety of CAL applications involving a variety of programming languages. Experience of the panel as a whole involves applications of all of the programming languages listed in Fig. 4 as well as applications involving translation from one language to another.

The functional specification (2) produced by the working panel has recently been approved by the Associate Committee for publication in order to receive critical comments in respect to its anticipated implementation as a national standard.

It is expected that the report will be available for distribution in early July.

It is not the purpose of this paper to describe in detail the requirements of a CAL programming language, but some of the important general characteristics are worth considering briefly.

Of particular importance in any high level, or problem-oriented language, is that it provides effective access to the computer for the intended user. It is not sufficient to incorporate in a language very powerful features, but implement them in such a way that the user has to be a mental contortionist to apply these features. The features of the language should be readily accessible to the novice user, but at the same time, the language should be sufficiently rich that it does not limit the experienced programmer.

For CAL applications, extensive features are required in the language for accepting and processing user input and for controlling the presentation of information. Glaser, Ramage and Lipson have carried out an extensive study of the interface requirements for teaching elementary mathematics, reading and science.(3) Their findings indicate the need for extensive input-output facilities in all three of these areas. Any CAL language must take into account these requirements dictated by subject matter, but must also give consideration to user requirements. The user may be, for example, a young child who cannot read, or he may have a perceptual handicap requiring special input-output facilities. Thus the language must be capable of coping with a wide range of input-output devices and provide the user with effective access to these devices in a variety of applications.

The language must provide sufficient facilities for the gathering and analysis of data which is generated as a result of applications. Such data may be used for on-line adjustment of programme logic, or they may be used for off-line analysis of student and course performance to provide measures for evaluating and improving programme materials.

Very much of importance is the requirement that the language be structured in such a way as to promote built in documentation. It is a fact of life that many programmes are written with the intention of adding explanatory comments at a later date which never arrives. The result is in many cases a programme whose logic is readily understood only by the original programmer and even he may have difficulties if he has not been working with the programme for some time. However, some programming languages are structured in such a way as to promote built in documentation, which, with the addition of relatively few explanatory comments results in a programme readily understood by potential users.

A requirement for Canadian applications is that the language have a structure such that programmes can be written in both English and French. This implies equivalent commands in the two languages, but also coded character sets which cater to the special requirements of information interchange in both languages. This is not to imply that there should be automatic translation of course materials, but it should be possible to run a course written in French on a system which is primarily used for courses written in

English and vice versa and all input processing functions should be valid.

If the programming language is to continue to remain valid, it must be capable of being updated to satisfy new user requirements and to permit advantage to be taken of new technological developments. Such updating must be introduced in a manner so as not to invalidate materials previously prepared using the language.

These are just a few of the general characteristics required of a programming language for CAL applications, but it is hoped that they are sufficient to indicate the complexity of the language and the problems which will be involved in its development.

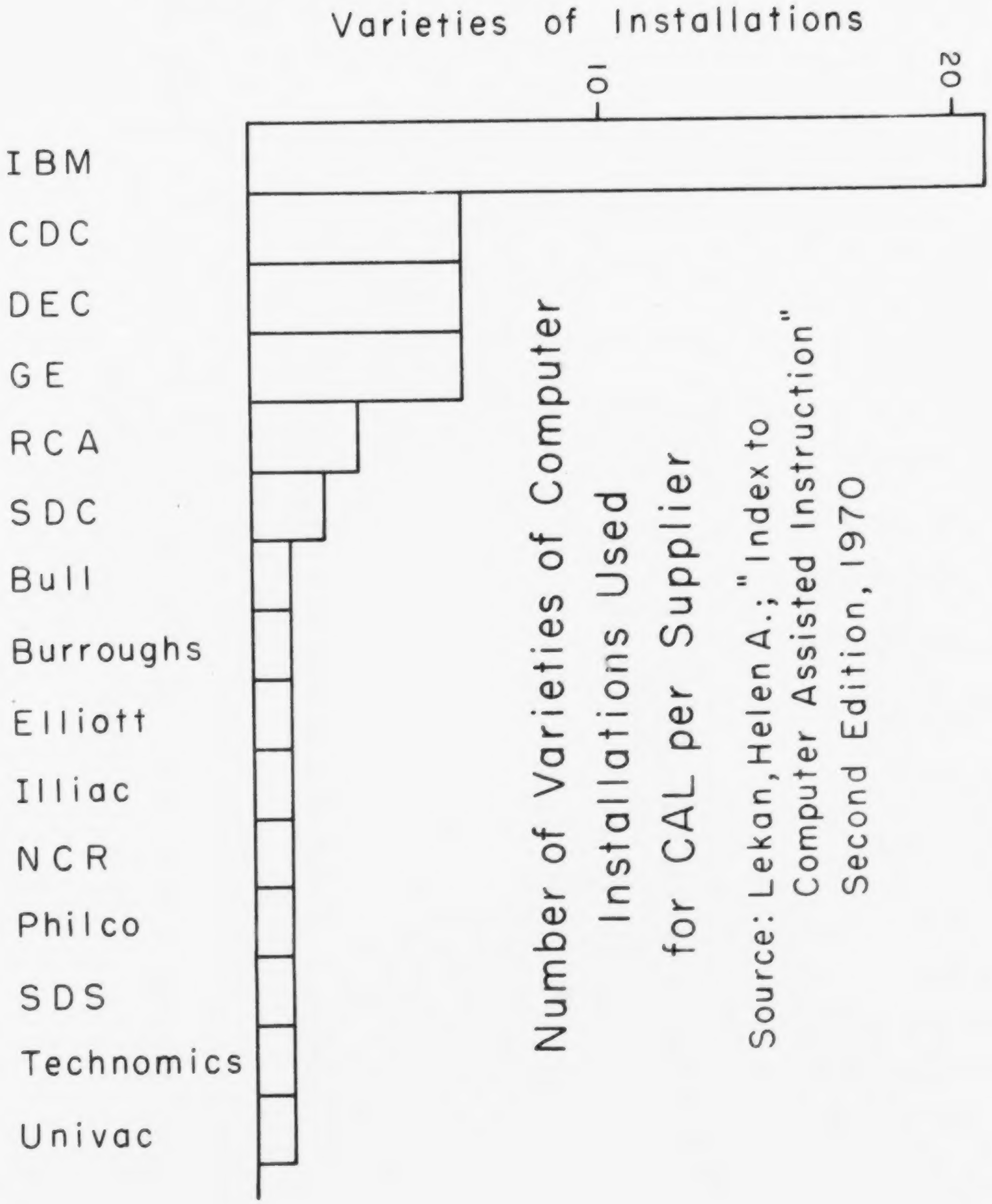
The functional specification for a Common CAL Language which has been prepared by the Associate Committee's working panel is but a first step in the process of arriving at a suitable language standard. The next step is the preparation of a detailed specification to define the fine structure of the language. This detailed specification should be based in whole or in part, on a currently existing language if one exists to satisfy many of the requirements defined in the functional specification. There is no point in developing the equivalent of what already exists. The detailed specification must be followed by implementation of the language at a number of centres. Until this implementation stage is reached, one cannot say that a standard truly exists, since any standard has very little meaning to the user if he cannot apply it and is forced into using available alternatives. The implementation and maintenance of a common programming language for CAL applications will require the continuing cooperation between the various users, equipment suppliers and research groups. Without such cooperation, there can be little hope of achieving a meaningful standard.

REFERENCES

1. Lekan, Helen A. - "Index to Computer Assisted Instruction", Second edition, Sterling Institute, Boston, Mass. 1970.

2. - "A Functional Specification for a Programming Language for Computer Aided Learning Applications" National Research Council of Canada Associate Committee on Instructional Technology (to be published), Ottawa, Canada, 1972.

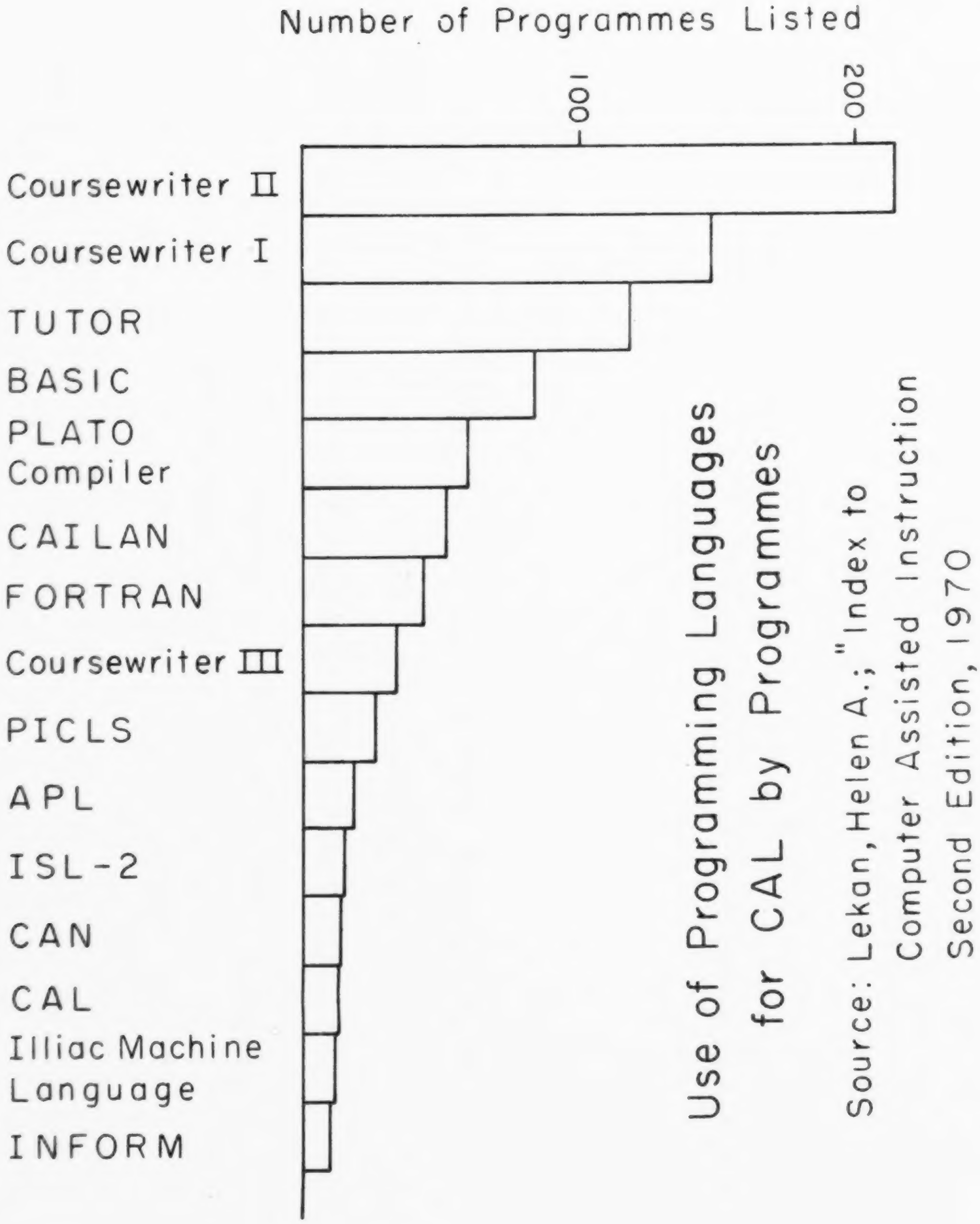
3. Glaser, R., Ramage, W. W. and Lipson, J. I. - "The Interface Between Student and Subject Matter" Learning Research and Development Center, University of Pittsburgh, Pa. 1964.



Number of Varieties of Computer
Installations Used
for CAL per Supplier

Source: Lekan, Helen A.; "Index to
Computer Assisted Instruction"
Second Edition, 1970

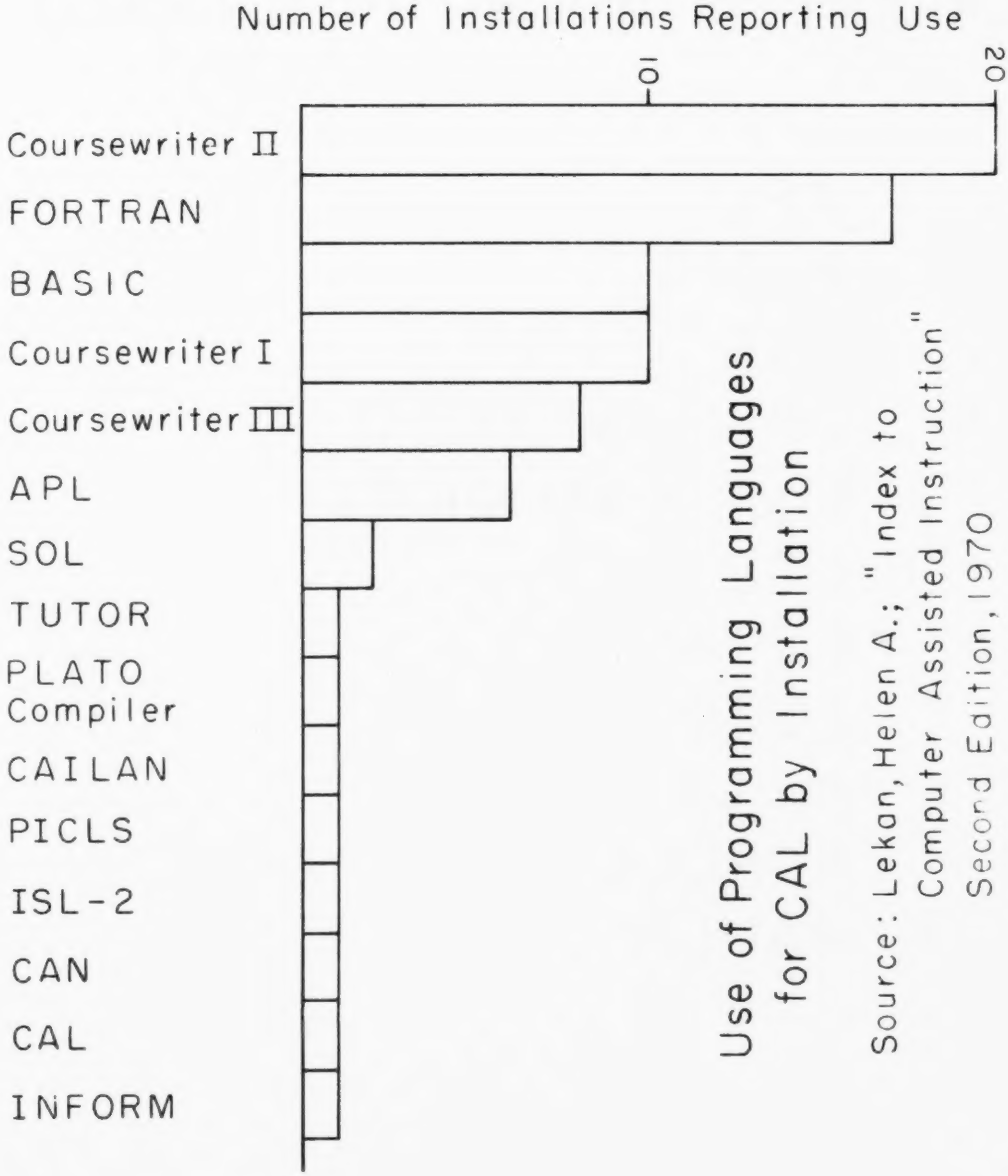
Figure 1



Use of Programming Languages
for CAL by Programmes

Source: Lekan, Helen A.; "Index to
Computer Assisted Instruction
Second Edition, 1970

Figure 2



Use of Programming Languages for CAL by Installation

Source: Lekan, Helen A.; "Index to Computer Assisted Instruction" Second Edition, 1970

Figure 3

CAL PROGRAMMING LANGUAGES CURRENTLY IN USE IN CANADA

APL
BASIC
CAN
COURSEWRITER II
COURSEWRITER III
ETL
FOCAL
LOGO
MULE
SCROLL
VAULT

Figure 4

Interactive Programming Languages Adapted for Specific Instructional Uses

Karl L. Zinn
University of Michigan

Some of the text is adapted from presentations at the OECD Seminar on Computer Sciences in Secondary Education held at Serves, France, March 1970, and the First International Conference on Man-Machine Interaction, Teddington, England, September 1970. Ideas proposed in the 1970 statements are repeated here as realities of instructional software on the Michigan Terminal System in 1972.

SESSION 10
ROOM/SALLE 148

INTERACTIVE PROGRAMMING LANGUAGES ADAPTED FOR SPECIFIC INSTRUCTIONAL USES

By/Par: Karl L. Zinn,
Centre for Learning & Teaching,
University of Michigan.

The essential contribution of interactive computer use involves responsiveness: a response to the efforts of the user which is relevant to his needs and background; provisions for encouraging effective response of the user to the system effort. During preparation and execution of a program, responsiveness provides the casual and infrequent user special benefits of procedure writing and testing of algorithms.

After a brief review of a dozen interactive programming languages, an instructional environment will be described in which the teacher or lesson designer builds upon task-relevant features and is able to open and close student access at various levels of use. For example, the student learning from a computer-based game or simulation can become the "instructor" when setting up a new exercise for another student, and the instructor can become a "researcher" exploring the models on which the learning games were based.

LE LANGAGE INTERACTIF DE LA PROGRAMMATION ADAPTÉE À DES UTILISATIONS PRÉCISES DANS L'ENSEIGNEMENT

La contribution essentielle de l'utilisation de l'ordinateur interactif implique la réponse: une réponse aux efforts de l'usager pertinente à ses besoins et à son expérience; des dispositions pour l'encouragement de la réponse efficace de l'usager à l'effort du système. Pendant la préparation et l'exécution d'un programme, la réponse fournit à l'usager fortuit, qui ne s'en sert qu'à l'occasion, des avantages spéciaux d'écriture de procédé et d'essai des algorithmes.

Après un bref examen d'une douzaine de langages de programmation interactifs, on décrira un environnement enseignant où le professeur ou l'auteur de la leçon, base sa conception sur des caractéristiques pertinentes à la tâche et peut ouvrir ou fermer l'accès de l'étudiant à des niveaux d'usage différents. Par exemple, un étudiant qui est en train d'apprendre au moyen d'un jeu ou d'une simulation à base d'ordinateur, peut devenir le "professeur" au moment d'établir un nouvel exercice pour un autre étudiant et le professeur peut devenir le "chercheur" examinant les modèles formant la base des jeux.

The essential contribution of interactive computing involves responsiveness: a computer should respond to the efforts of a user in a way which serves his goals and recognizes his needs and background; and computing procedures should encourage effective response of a user to the computing done for him. During preparation and execution of a program, responsiveness provides the casual and infrequent user special advantages for writing procedures and testing algorithms.

After a brief review of a dozen interactive programming languages, an instructional environment is described in which the teacher or lesson designer builds upon task-relevant features and is able to open and close student access at various levels of use.

Availability of Languages

Interactive language processors have a short history, although console use of small machines kept alive the idea of interactive tracing and debugging while large computation centers were completely dependent on batch facilities.

It has been about eight years since RAND Corporation made JOSS (Shaw, 1964; Bryan and Smith, 1967) available to its scientists. Initially it was an experiment in "open-shop" service for simple computing needs and used an old computer no longer considered of value for research users. It brought convenient computing power into professional offices, and was characterized in the mid-1960's as a major fringe benefit of working for RAND Corporation. Ready acceptance by users and favorable impact on working habits have been characteristic of conversational computing.

TELCOMP, a derivative of JOSS prepared at Bolt Beranek and Newman (Meyer, 1966), has been used with considerable success by elementary school children for individual exercises in mathematics and sciences. TELCOMP and its descendats STRGCOMP (Bolt Beranek and Newman, 1968) and ISRCOMP (Feurzeig and Papert, 1969) continue to be used for programmer training at all levels, and for student exploration of mathematical concepts.

LOGO found its initial success in teaching fundamentals of mathematics to children about 12 years of age (Feurzeig and Papert, 1969) and the current emphasis of its originators is on teaching young children to think (Papert, 1970; Papert and Solomon, 1972). It is now being adopted by a number of other research projects for development of environments for learning.

BASIC built a reputation on early success at Dartmouth College (Kemeny and Kurtz, 1967) and throughout GE time-sharing service centers. The name and most of the language features were adopted by other manufacturers and time-sharing services, and now standardization has become a concern (Ogden, 1971).

APL (Falkoff and Iverson, 1968), built upon the notation proposed by Iverson (1962) in his book A Programming Language, has been applied by different users to various tasks, including student problem solving and simulation. For at least some computer-assisted instruction projects it has replaced COURSEWRITER for programming tutorial as well as simulation exercises.

Much of the enthusiasm for conversational computing languages may relate to non-essential features; the quick response and understandable diagnostics of these on-line systems have been provided in batch systems. BASIC is often used as a remote entry batch system. The Illinois Institute of Technology has for some time serviced a network of school and college users with good results at very low cost using IITRAN (Dewar and others, 1969) for remote-entry batch (along with a desk-calculator-type interpreter called CALCTRAN).

Now that commercial services and small dedicated systems are being offered to (and purchased by) schools and colleges, it becomes increasingly important to isolate the essential contributions of interactive programming languages, and to determine effective cost conditions.

Computer systems dedicated to instruction uses in a tutorial mode cost between $2 to $12 per user hour at a console (or user station). The estimates depend on assumptions about the number of simultaneous users which can be service with adequate response time, the number of effective hours of use per day or per year, the years of operation of the system, the amortization of software costs, etc. Convenient means for presenting static, graphic images generated in real time by the computer can add considerably to the cost, although graphics will be much less expensive in the near future.

Small systems dedicated to problem solving use of a simple interactive programming language cost from 25¢ to $8 per user hour at the terminal. Again the values depend on the expectations of the buyer for effective use of the system, that is, number of simultaneous users which can be supported, effective hours of use, amortization, etc. Commercial time-sharing services range from $5 to $35 per hour, depending in part on the particular means of accounting for elapsed time connected to the computer system, central processor time, channel use, and storage. Incidentally, a user with access to two or more systems sometimes can extend his computing dollars by matching his problem demands to billing techniques. For example, he would run compute-bound jobs on the system billing only for time connected, and carry on conversations making heavy use of the terminal and input-output channel on the one emphasizing central processor time in the billing. Of course the differential advantage will be reduced as many users take advantage of the differences in billing procedures from one company to another.

Desirable Features for Interactive Use

Part of the success of interactive programming can be explained by the ease with which the languages are learned and initially used, whether in batch or conversational mode. FORTRAN compilers have presented a difficult hurdle for the uninitiated; a new user's first attempts to specify input lists, output formats and conditional branching can be frustrating. JOSS has a small number of statements and considerable consistency throughout; this simplicity (or elegance) facilitates recall and use of statements in the language and is impressive when seen in interactive mode. Much of the advantage could be achieved with a similar convenience available readily through a batch-oriented system having a response time which is suitable for the learning and practice task involved.

The essential contribution of the interactive mode of work must involve responsiveness of the system: relevant reaction to the efforts of the user; and provisions for encouraging effective response of the user to the system effort. Responsiveness of a processor during preparation and execution of a program is a key for the casual and infrequent user to unlock the special benefits of procedure writing and testing of algorithms. For example, the occasional user is well advised to try an idea (an expression, an untested subprocedure, or a new output format), even though only partly formulated, and let the computer show him how much can be processed and what the results might be. Furthermore the processor can "guess" at what the user intended, begin executing the instructions, and accept an interruption and clarification as soon as the user recognizes that the procedure should be changed.

When diagnostics are provided at the moment when needed and backed up by references to readily available literature, one can relieve the user of unnecessary concern for detail of the means to describe his procedure; one hopes he will give more attention to solving the problem. A shorter elapsed time between problem definition and solution, and the time saving attritutable to continuous working sessions provide a bonus.

Interactive programming languages incorporate aids for program testing in a very natural way. The same statements with which stored programs are written can be used as direct commands to print the values of selected variables to see what went wrong, assign new values to test other parts of the procedure, and resume execution with any line or segment of the program.

On-line conversational use of computers is almost certain to be more costly than less glamorous means of access to information processing aids. However, the occasional user of computers is more likely to benfit from the interactive mode of operation than experienced or frequent users. Programming languages and learning exercises should be slected to take advantage of opportunities for the infrequent learner-user to carry on a dialogue with the system. His attitude toward computers and their uses in his discipline may at times be as important as his particular accomplishments with the specific content skills to be learned.

Authors of computer-based learning exercises, although their primary task is instruction and not programming, probably benefit from all of the advantages listed above. The computation facilities of the interactive languages are useful in many computer-based learning exercises, and convenient computation is conspiciously absent from nearly all languages specially designed for computer-assisted instruction. When an author uses an interactive language the program is more accessible to the student. This increases the opportunity for a student to become involved in the redesign of an exercise, and this is especially important when the curriculum writer has incorporated mathematical models or simulations.

Extension into Other Disciplines

In order to build a proper problem-oriented language for many subject experts to use in a variety of tasks such as building models and embedding them in pedagogical situations, the language itself ought to be extendable in some way through macros, subroutines, or statement definition capability.

One suitable means for extending a language also provides additional capability for servicing a variety of users, and shifting a user among different modes. Additional features should be implemented as another level of operation in the same language system, and facility should be provided for opening or closing access to the various levels defined. For example, the student learning from a computer-based game or simulation can become the "instructor" when setting up a new exercise for another student; the author can become a researcher exploring models on which his learning games are based; even the applications programmer can become a "systems" programmer, if he can move from the level of programming in a given language to the design of new language features.

The system designer (or an experienced programmer) can on the basis of discussion with a subject expert add to the language those statement types which service the particular needs of the problem. The subject expert should find this second-level notation convenient for building models and for embedding them in a suitable learning situation. Of course he may delegate this task to a writing assistant or programmer. The student user of this system works at a third level, interacting with the model at first only through the abstract simulation as a management game to be played. The lesson designer provides the student with a means to "unlock" the features of the processor by which the model may be manipulated in simple ways, and the student may explore its parameters and perhaps arithmetic expressions and logical conditions. At this point a student will be permitted to move down to the 2nd level, that of the author or subject expert; here he may design entire models or simulated situations. Whenever the task-oriented features of the language are not adequate to the model building and testing, the experienced programmer is again called in to extend these capabilities to meet the needs of the discipline-oriented users.

This conception can be represented in part by an arrangment of successive shells with permeable boundaries. The system configuration lies at the center of the sphere, and the system programmers build up a shell around it to make useful functional capabilities accessible to applications programmers. They in turn produce the routines and packaged procedures which provide convenience and power for the discipline-oriented user, who may then adapt these task-oriented procedures for convenient use by students in suitable pedagogical content. The outer three or four levels are the ones of primary concern in this paper: instructional use; discipline-related packages and applications programming; and modification of the processor and system features.

Because of the variety of instructional strategies, intended learning outcomes and writing habits, it is not feasible to include all the desired aspects of this multi-level situation in a single language. More appropriately one works within a system which provides the necessary building blocks and the means for linking them together into individualized versions of a processor or package of processors.

Desirable Properties of an Instructional Programming Language and Operating System

The actual components in the operating system and processor are important to this discussion only as a general user might see them. Nine points follow.

1. Rapid response to trivial requests. For example, file manipulation, simple calculations or operations should be performed with no noticeable delay.
2. Retention of programs and data for the convenience of the user. During explorative execution of tasks the user should be able to return to any earlier state; when he concludes a working session he should be able to save the program and data for restarting (in the same state) at a later time.
3. Simplicity, conciseness and precision in notation. Typing a large number of characters is annoying to any serious user and leads to errors. The medium of communication between man and machine should be straightforward.
4. A suitable vehicle for programming. A sound design should encourage the user to derive a problem solution in terms which produce effective programs.
5. Notation to record algorithms clearly. A medium of communication among program writers and users should be possible.
6. Convenience in a variety of applications. The value of the language is increased by the number of diverse applications a single user may pursue. Although the designer cannot anticipate everything a user might want, he should try to provide the elements from which anything of which the system configuration is capable can be called upon in a convenient way.
7. Complete determination of the environment. The designer of a working environment must be able to specify handling of all events during interaction. He must be able to "recover" in spite of student errors, and even analyze those errors in order to restart in a useful state.
8. Control of the working environment with "protection." The user is in control or can recover it from the computer, although the system protects the user from errors such as accidental elimination of file records.
9. Relevance of the environment for work. The services of the computer installation should be available in some reasonable manner, e.g., acquiring various storage media, accessing one or another processor language, linking processors through common data files, editing, tracing, testing, etc.

The dialogue between teacher and student (or between scholar and data) is central to the learning environment; conventions for describing input and output should be convenient. Various devices must be accessed, including special-purpose terminal equipment for science experiments or artistic composition, and secondary storage such as disk files to hold large data bases on the topic being studied or to record detailed records of student performance. Rapid data rates are required for topics requiring extensive and dynamic graphics. Format conventions should allow options specified by the lesson designer, e.g., to control the position of the carriage or cursor both for reading and writing, and variables by which formats are generated dynamically to extend the designer's control of the working environment on the typewriter page or display screen. Complexity of input-output processing should be determined by the user's intentions and his competence with the language and system.

The manipulation of files is common in a problem solving environment. File-handling statements should be executable in the same way an assignment statement is. Essential functions are: listing; manipulation (searching and editing); designation of access (read-only, shared, write-only); and determination of type (sequential, indexed). Interrupts are an important consideration, with some handled automatically for convenience, but others left to the control of the designer of the working environment. He will interrupt to help the user recover from language and system features not yet understood, and perhaps interrupt at other times on the basis of how much time the user has spent or some other condition. Control of interrupts also allows the system to respond to student users directly for example, for evaluating expressions.

Programming a machine to respond usefully to a student's constructed command or comment requires a great deal of data handling. A facility for describing and manipulating a variety of data structures is essential if the language is to be flexible enough to encompass a variety of applications. Presently languages have only primitive methods for the description of data; current means for manipulation of data structures are also quite crude. Contributions of computer science to the study of different data structures have not made implementations any more available to the non-professional. Nevertheless much that is done in computer-based learning environments requires transforming and accessing information which entails a data description indicating the interrelationships between atomic elements. Because reasonable methods for making such a description are not available, the non-professional finds many tasks cannot be described for the computer. A more powerful data manipulation and descriptive facility could alleviate the problem.

Some way of extending a language is needed since anticipation of all demands will not be possible, and some will be so specialized that they should not take resources for development and maintenance of the basic processor used by all. Various means are available: single- or multiple-line definitions of subroutines; macros with parameters; linkage to other processors; and definition of new operators or statement types. Extensible languages are much discussed but little understood, and it appears that experimental languages may be extended only by the system programmers because of the many intricacies in such systems. An easily modifiable language might be achieved more by careful attention to the organization of the implementation of the language.

The implementation of all these features in a single language which can realistically be used in computer-based learning environments will require different techniques for language organization and implementation. New techniques will also have to be applied, especially in data manipulation and description.

Summary of General Considerations for Innovative Developments

The major consideration at this stage in development of facilities and procedures must be flexibility. For example, implementation of one or more processors and their linking mechanisms within a general-purpose system will enable a systems group to explore new ideas in the same environment with initial users, and then incorporate those which show promise in the preliminary uses and develop them further as use justifies.

The underlying assumption to be recognized is that different users and uses have different needs. Is the user learning to program, computing a statistic, processing a text, running a simulation, building a model, drafting a learning exercise, etc.? In an operational setting, a sub-system, language subset, or applications package may be tuned to the needs of specific users; student, instructor and lesson designer each require a different kind of support from the system and a different style of training and interaction with the system.

Interactive mode considerations are especially important for the infrequent users. Four statements are made as working hypotheses to be demonstrated by studies at the University of Michigan.

1. An immediate and responsive reply from the system will encourage efficient and effective use of the computer as a learning tool. Whenever the student-user doesn't know where he is in the system or learning exercise and cannot determine what to do next he should be able to get a useful reply by questioning the system, and he should be able to determine a suitable place to resume the exercise.

2. Rules and conventions should be easy to learn and later recall when working at the rapid pace of conversational interaction. The features of simplicity and interal consistency which are important in a batch-oriented system for student uses become especially powerful when the user works in interactive mode, trying, testing and constructing without reference to manuals, or other assistance.

3. A comfortable dialogue is possible in which the computer processor neither dominates nor leaves the user "in the dark." The system should be engineered so that the user can do for himself those things he does better than the machine, and the machine can serve the user in its area of strength. Machines attend to detail, work repetitious tasks rapidly and without error or complaint, access large stores of information without memory failure, etc. People attend to larger structures and patterns, set direction for promising investigation, and determine values.

4. Flexibility during a working session will encourage a user to test out an idea, compare different versions of a subroutine, save procedures for future use, and select the most economical as well as the most convenient of alternates. The learner-user should feel encouraged to test tentative ideas and try out possibilities, knowing the system will not only permit such explorations, but will help them to be successful. A suitable language processor will keep track of loose ends while the user is sketching in ideas, accept details later on, and provide immediate and interpretable reply when the user's instructions are ambiguous or incomplete.

Although the problems presented here appear to reside in the inadequacies of the computer and information processing systems, for the most part they will not be solved by development of new computer software and hardware alone. Solutions will be achieved only after scholars and educators examine current methods of teaching and research in each discipline and make adaptations for the new age of informatics. Developments in operating systems and language which facilitate a problem orientation of software will encourage the expert but not solve his problems for him. Total reevaluation of the instructional process and the desired objectives for technicians and scholars are in order if the computer and information processing are to find a suitable place in the educational and professional environment.

A List of Languages, Including Sources for Current Information

ACME: Advanced Computer for Medical Research. Real-Time Computation Facility, Stanford University Medical School, Stanford, California 94305

ALCOM: Applied Logic Computing. Applied Logic Corporation, One Palmer Square, Princeton, New Jersey, 08540.

APL: A Programming Language. IBM Scientific Center, 3401 Market Street, Philadelphia, Pennsylvania, 19104. Commercially available.

BASIC: Beginner's All-purpose Symbolic Instruction Code. Computer Center, Dartmouth College, Hanover, New Hampshire, 03755. Commercially available.

BRUIN: Brown University Interactive Language. Brown University, Computing Center, Princeton, New Jersey, 08540.

CAL: Conversational Algebraic Language. Computer Center, University of California, Berkeley, California, 94720. Commercially available.

CITRAN and REL: CIT Translator; and Readily Extensible Language. Computing Center, California Institute of Technology, Pasadena, California, 91109.

FOCAL: Formulating On-Line Calculations in Algebraic Language. Digital Equipment Corporation, Maynard, Massachusetts, 01754.

IITRAN and CALCTRAN: IIT Translator; and Calculating Translator. Computation Center, Illinois Institute of Technology, Chicago, Illinois, 60616.

ISIS: Irvine Symbolic Interpretive System. Computer Facility, University of California, Irvine, California, 92664.

JOSS: JOHNNIAC Open-Shop System. Computer Sciences Department, RAND Corporation, 1700 Main Street, Santa Monica, California, 90406.

LCC: Language for Conversational Computing. Computation Center, University of Pittsburgh, Pittsburgh, Pennsylvania, 15213.

LOGO. Department of Educational Technology, Bolt Beranek and Newmen, 50 Moulton Street, Cambridge, Massachusetts, 02138.

PIL: Pittsburgh Interpretive Language. Computing Center, University of Pittsburgh, Pittsburgh, Pennsylvania, 15213.

POP-2. Department of Machine Intelligence, University of Edinburgh, Scotland.

QUIKTRAN: "Quick" FORTRAN. Information Marketing Publications, International Business Machines, Montereyand Cottle Roads, San Jose, California, 95113.

RUSH: Remote User Shared Hardware. Allan-Babcock Computing, Los Angeles, California, 90067.

TELCOMP, STRCOMP and ISRCOMP. Bolt Beranek and Newman, 50 Moulton Street, Cambridge, Massachusetts, 02138.

TINT. System Development Corporation, Santa Monica, California, 90406.

Bibliography

Boettner, Donald W. *Command (Job Control) Languages for General Purpose Computing Systems*. Computing Center, University of Michigan, Ann Arbor, June 1969.

Bolt Beranek and Newman, Inc. *STRGCOMP Manual* (an extension of TELCOMP). Cambridge, Massachusetts, April 1968.

Branden, Helen V. and William A. Wulf. "The Implementation of a BASIC System in a Multiprogramming Environment." *Communications of the ACM*. Vol. 11 (October 1968), pp. 688-92.

Brietbard, Gary and Gio Widerhold. "The ACME Compiler." *Proceedings of the IFIP Congress 1968*. Amsterdam: North-Holland, 1969.

Bryan, G.E. and J.W. Smith. *JOSS Language: Apercu and Precis*, RM 5377-PR/2. Rand Corporation, Santa Monica, California, August 1967.

Burstall, R.M., J.S. Collins and R.J. Popplestone. *POP-2 Papers: Introduction and Reference Manual*. Edinburgh: Oliver and Boyd, 1968.

Christensen, Carlos and Christopher J. Shaw, Editors. *Proceedings of the Extensible Languages Symposium*. SIGPLAN Notices (entire issue), Vol. 4, No. 8 (August 1969).

Dewar, Robert B.K., Ronald R. Hochspring and William S. Worley. *The IITRAN Programming Language*. Illinois Institute of Technology, Chicago, 1969

Falkoff, A.D. and K.E. Iverson. "The APL/360 Terminal System." *ACM Symposium on Experimental Systems for Applied Mathematics* Academic Press, 1968.

Feurzeig, Wallace. *Computer-Systems for Teaching Complex Concepts*, Report No. 1742. Bolt Beranek and Newman, Inc., Cambridge, Massachusetts November 1969.

Feurzeig, Wallace, Seymour A. Papert, M. Bloom, Robert Grant and Cynthia Soloman. *Programming Languages as a Conceptual Framework for Teaching Mathematics*, Report No. 1889. Bolt Beranek and Newman, Inc., Cambridge, Massachusetts, November 1969.

Hargraves, Robert F., Jr. and Andrew G. Stephenson. "Design Considerations for an Educational Time-Sharing System." *1969 Spring Joint Computer Conference*. Montvale, New Jersey: AFIPS Press, 1969.

Iverson, K.E. *A Programming Language*. New York: Wiley, 1962

Kemeny, John J. and Thomas E. Kurtz. *The Dartmouth Time-Sharing Computer System*. Dartmouth College, Hanover, New Hampshire, June 1967.

Kennedy, Phyllis R. *The TINT User's Guide*. System Development Corporation, Santa Monica, California, July 1965.

Lampson, B.W., W.W. Lichtenberger and M.W. Pirtle. "A User Machine in a Time-Sharing System." *Proceedings of the IEEE*, December 1966, pp. 1766-1774.

Mitchell, I.G., A.J. Perlis, and H.R. Van Zoeren. "LC^2: A Language for Conversational Computing." *Interactive Systems for Applied Experimental Mathematics*. Edited by Lkerer and Reinfields. 1968. Pp. 203-213.

Munck, Robert G. "Meeting the Computation Requirements of the University: The Brown University Interactive Language." *Proceedings of 24th National Conference*. Association of Computing Machinery, New York, 1969. Pp. 665-671.

Odgen, Jerry L. "The Case Against BASIC." DATAMATION, September 1, 1971. See also comments in SIGCUE Bulletin, Vol. 5, #5, October 1971, pp. 217-226.

Papert, Seymour. "Teaching Children Thinking." *World Conference on Computer Education*, B. Scheepmaker and K. Zinn (Eds.), International Federation for Information Processing, Amsterdam, 1970.

Papert, Seymour and Cynthia Solomon. "Twenty Things to Do With a Computer." Educational Technology, pp9-18, Vol. 12, #4, April 1972.

Pittsburgh, The University of. *PIL/L Pitt Interpretive Languages for the IBM System/360 Model 50*. Pittsburgh, March 1968.

Shaw, J.C. *JOSS: A Designer's View of an Experimental On-Line Computing*, P-2922. RAND Corporation, Santa Monica, California, August, 1964.

Thompson, F.B., P.C. Lockemann, B. Dostert and R.S. Deverill. "REL: A Rapidly Extensible Language System." Proceedings of 24th National Conference. Association for Computing Machinery, New York, 1969.

Wirth, Niklaus and C.A.R. Hoare. "A Contribution to the Development of ALGOL." Communications of the ACM, Vol. 9 (June 1966), pp. 413-431.

Zinn, Karl L. A Comparative Study of Languages for Programming Interactive Use of Computers in Instruction. EDUCOM, Boston, Massachusetts, February 1969.

Zinn, Karl L. Requirements for Programming Languages in Computer-based Instructional Systems. Centre for Educational Research and Innovation, OECD, Paris, 1971.

Zinn, Karl L. "A Personal View of Computing Software for Use in the Instructional Process." International Computing Symposium Proceedings, ACM, 1972, pp. 402-412.

COMPUTER-AIDED DESIGN INSTRUCTION
A COMBINATION OF DO-IT-YOURSELF PROGRAMMING AND COMPUTER PACKAGE USE

by

Wolfgang Bitterlich
Applied Mathematics Department
Nova Scotia Technical College

and

Dimitri Procos
School of Architecture
Nova Scotia Technical College

SESSION 12A
ROOM/SALLE 143

COMPUTER-AIDED DESIGN INSTRUCTION
A COMBINATION OF DO-IT-YOURSELF PROGRAMMING AND COMPUTER PACKAGE USE

By/Par: W. Bitterlich, D. Procos
Nova Scotia Technical College

This describes a course aimed at strengthening those aspects of the design process which are frequently bypassed by most architectural students in their school design work; usable problem analysis, alternative design development and design evaluation.

This objective is achieved mainly by introducing the students to a number of computer programming packages which can be applied by them directly to the current design problems, with a minimum of programming knowledge. At the same time the student will be taught rudimentary programming in the Fortran language to insure greater comprehension of the above programs and, especially, to give interested students the tools to revise them and to develop new ones. Programming instruction is intended to avoid programming exercises that do not have direct relevance to architectural problems and will instead use simple architectural problems leading to short (10-30 statement) computer programs.

The course is being given for the first time at the Nova Scotia Technical College to a group of Architectural students by the authors of this paper, one of whom is an Applied Mathematician who will deal with programming instruction, while the other, an Architect, will formulate software relevant to design problems. Positive reinforcement techniques will be applied to course grading in that higher grades will depend on the willingness of the students to make use of the packages as well as on the quality of their own programming performance.

Initial design studio feedback on the success of the course will be available when the paper is delivered at the symposium.

LA FORMATION DES ARCHITECTES À L'AIDE DE L'ORDINATEUR
L'ÉLÈVE-ARCHITECTE FAIT SES PROGRAMMES OU EN CHOISIT PARMI CEUX ÉTABLIS À SON INTENTION

Dans cette communication, nous décrivons un cours dont le but est de renforcer ces aspects de la conception qui sont souvent délaissés par la plupart des élèves-architectes dans leurs études: l'analyse des problèmes utilisable, les différentes solutions possibles et leur évaluation.

On atteint ce but principalement en présentant à l'élève un certain nombre de programmes tout faits dont il peut se servir en ayant des connaissances minimales de programmation. En même temps, on apprendra à l'étudiant les éléments de base du langage Fortran pour qu'il ait une meilleure compréhension de ces programmes et, surtout, pour lui fournir des outils lui permettant de modifier et de développer d'autres programmes. L'instruction vise à éviter les exercices de programmation n'ayant pas un rapport direct avec les problèmes d'architecture et à présenter des problèmes architecturaux conduisant à de courts programmes de 10 à 30 énoncés.

Ce cours est donné pour la première fois au Collège technique de la Nouvelle-Écosse à un groupe d'élèves-architectes par les auteurs de cette communication, dont l'un est spécialiste des mathématiques appliquées et qui enseignera la programmation, tandis que l'autre, qui est architecte, s'occupera de la programmation se rapportant aux problèmes de la conception. Une technique de renforcement s'appliquera à la classification de cours, de sorte que les élèves atteindront les niveaux plus élevés en fonction de l'effort qu'ils feront pour utiliser les programmes et de la qualité de leur propre rendement.

Les premiers résultats sur le succès de ce projet seront disponibles au moment de la Conférence.

"We kept saying to the computer people yes it's fine for production drawings or for estimating once you have the basic design established, but how do you get to the point? They said to tell them how we design and they'd work something out. We told them something different every day depending on what we were doing that day. It was a case of the analysts showing architects what analysts could do. We had to show them what architects could do."

J. R. Myer and R. I. Krauss
Center for Building Research
Mass. Inst. of Technology

One of the main problems in using the computer for the teaching of "rationalized" design processes is getting around the danger of losing the student in the process of teaching him the rudiments of computer usage. There are a number of danger points: (1) the examples commonly used in programming instruction (say the teaching of the Fortran language) are often irrelevant to architecture. They tend to turn-off all but the most patient students. (2) The architectural problems which the student is supposed to solve with his newly acquired resources necessarily have to be kept simple. The student argues rightly that he already operates at a much more sophisticated level or that he can do a much better job of tackling these problems manually. (3) Programming and computer expertise in general is cumulative. It becomes must useful at the end of the learning process when it is perhaps least needed, since present computer applications seem to be most useful in the analytic stage of the design process (the evaluative stage which does occur at the end of a design program would also lend itself to computer treatment but after all evaluation is heavily dependent on the initial analysis). (4) Programs that are available to the student in the latter parts of a design project when he has the expertise to work with them often depend on the input of data which are beyond the student's data collection capabilities.

One solution to the problem would be to cut out programming from the architect's repertoire and to rely exclusively on packaged programs. But close observation proves this not to really be a solution: as the quote at the beginning of this paper implies, an architect should have at least a grasp of what a computer can do for him and what is beyond its scope in the present or near future. He should have some understanding of how the computer approach to problems differs from traditional design methods and, in case he wants to make use of these new possibilities, he has to communicate with systems people. This cooperation will be easier if the architect can express to the analyst what he wants done. Both have to meet somewhere in between.

A more realistic solution then would be to introduce a broad range of computer applications, all the time making sure that Fortran is not being taught for architectural applications but rather that design techniques are being taught through the medium of Fortran. Essentially what we are after is using the computer for those things in the design process that students don't do or do badly.

We chose a three-pronged attack on the problem:

(a) The use of as architectural a vehicle as possible for teaching programming languages to avoid alienating the student during this the longest part of the course.

(b) The introduction of readily useable and reasonably useful computer packages for the early use of students, in order to "sell them" on the usefulness of the computer in their everyday work.

(c) The introduction of incentives for the students to use computerized processes in lieu of the usual intuitive ones.

These three components took the following detailed form:

(a) When the elements of Fortran were introduced, mathematics (as used in courses for engineers) were avoided whenever possible. Instead such areas were chosen which rely mostly on common sense and which might be of interest and use in the design process. This was intended to be carried out under two main topics:

(i) Simulation

Simulation can be introduced on a very basic level by starting with examples like throwing dice and playing games such as craps. Later, simple critical path networks with random job completion times, random search and queueing models can be introduced.

Parallel to that the student becomes familiar with the main elements of programming such as looping, branching, designing of flow charts. The parts requiring knowledge of non-trivial mathematics such as transformation of distributions are not to be taught but rather to be given as subprograms in the same way that an engineer does not actually learn how to program the square root function. The stress must be on designing the model and finding out the deterministic and stochastic relationships between its components.

Thus programming becomes similar or complementary to architectural design.

(ii) Use of Matrices

There is no reason for the student to learn how to multiply or invert matrices etc. Matrices should rather be used to describe, e.g. maps, relationship tabulations, and (mainly two-dimensional) space. Operations with matrices are restricted to multiplication by constants and addition. This is necessary and sufficient for combining parameters describing different aspects such as land cost, soil types or drainage. This type of matrix handling is extensively applied in some computer packages especially in space allocation programming.

(b) We saw how the sequence of programming can be gradually unravelled through the parallel development of simple simulation and matrix manipulation techniques. At the same time the student has to be given an early opportunity to make use of the new tool whose usefulness and potential he is barely beginning to perceive. This is done through computer packages made available to the student with explicit directions as to their use, some indication of their internal working and a clear idea of the kind of output that can be expected from them.

It should be pointed out that the packages themselves are not new. We used existing ones or wrote new ones based on somebody else's theory. This was not because we felt that existing software filled all architects' needs and could not be perfected. We simply felt that the greatest gap between program formulation and program use in computer aided design was due not so much to technical imperfections as to limited or superficial use of these programs by the users that they were planned for. This is why we decided to concentrate on the application of these programs in the students' design context. In fact we decided to try to interpret future successes or failures in the application of computer packages more in terms of the usefulness of a particular package to a particular student's needs at a particular stage in his or her design than in terms of the internal workings of the program. In short, we tried to approach the intuitive design process which is still the architectural student's major tool with a great amount of respect and circumspection. Specifically we chose the following programs:

1. For site selection in terms of distance requirements of the facility being designed for from certain given fixed points, we developed SPATLOC, a program derived from a paper entitled "Graphical Representation of a Matrix with Applications in Space Allocation" by C.E.S. Lindgren and C. Steinitz of the Centre for Environmental Studies at Harvard University.

2. For site selection in terms of optimizing a number of site characteristics we used CASAT, a program developed in PL I partly on the basis of a technique developed by Ian McHarg. This program presents the designer at various stages with alternatives to choose from. It gives the student the opportunity to fill "gaps' by formulating and programming more or less sophisticated selection criteria.

3. For the internal workings of a building in terms of arrivals and movements of users and material we have QUEUE, a program that simulates arrival and queuing.

4. For relating various spaces to each other in diagrammatic form we chose CLUMP, a program developed by W. J. Mitchell of U.C.L.A.

5. For a number of alternative attempts at inter-relating various spaces in actual physical plan form we used LOCAT, a program that prints out plan configurations without the use of a plotter.

It is seen that the programs described are developed along a certain sequence of operations which approximates the operations that a student goes through when tackling a design project intuitively. Firstly, site selection takes place on a basis of optimization of requirements and cumulative site characteristics. Secondly, a simulation is attempted of the kind of dynamic processes that will be happening when the facility is being used. Thirdly, the conceptual relationship of the various static spaces in the facility is determined and fourthly, a tentative physical design is generated.

(c) In trying to ensure that the students use the computer packages available to them, we soon realized that it would be unrealistic to use negative reinforcement techniques, ie, to tie a pass mark for the course (a mark of C) to the compulsory use of these packages. It would be too easy for the students to claim that their failure to use a certain package was due to flaws inherent in the package which they had been able to detect from the outset without having to run the actual program.

As a result we decided to switch to positive reinforcement, ie, to tie the use of computer packages to a system of rewards (in this case, marks above the passing C level) that would give a strong incentive for the student to try his hand at their use. It should be stressed again that we are not necessarily after successful applications that can demonstrably improve the intuitive design process; failures by the student in applying programs to his own design project would also be rewarded if they had clearly pinpointed the deficiency and if the student could cogently explain this deficiency. In fact a student could conceivably receive a further mark bonus for suggesting realistic additions or changes to the package that would show that he or she has understood the manual, even if the application of the revised program to his or her own design is again inconclusive. The actual marking system we devised would give a mark of A for two such attempts, in both cases regardless of the elegance or results of the attempt as long as there was an effort there.

At the bottom end of the marking scale we had to devise a way to determine the pass - fail component (the C mark). The procedure we decided to follow consisted of providing the students with a list of very simple problems that could be solved by short (10 to 30 word statement) programs, which would be within the programming capabilities the student would have acquired around

The middle of the course. Each student would have to write and successfully run two of these programs to reach a C mark. Alternatively the student could write his or her own problem and the corresponding program, based on current or projected needs within the design project, in which case one program would be enough for a pass mark.

Initial Feedback and Conclusions

In many ways student response to our course has worked out in the way we anticipated: a good number of students made use of the packaged programs quite early in the course so that, at the writing of this assessment (a little after the mid-term) about 50% of the students have had first-hand experience of a computer responding to their data and to their analysis of a design problem. Work on the programming component has been somewhat slower in coming but we are going on the assumption that a student who has already "reserved" a B or A grade will have a higher than usual motivation to do the pass-fail work necessary to confirm his higher standing.

Whether or not future developments bear out this assumption is of more than limited interest to us. A successful outcome will of course confirm the usefulness of teaching a body of knowledge against which there is often considerable resistance among architects and students, but it will also reflect favorably on the particular instructional methodology we used: that of separating work aimed at demonstrating student excellence from work aimed at showing mere student competence and of using the attraction of the first to overcome student resistance against the second. We feel that this approach is applicable to a wide variety of instructional situations. We also feel that computers, through their ability to "prepackage" much of the background material necessary for advanced learning experiences will be instrumental in bringing about its application.

The Use of Groups in Computer-Assisted Instruction

Glenn F. Cartwright

Department of Educational Psychology

The University of Alberta

Edmonton, Canada

A Paper Presented at the
Canadian Symposium on Instructional Technology,
The University of Calgary,
Calgary, Alberta
May 24-26, 1972

THE USE OF GROUPS IN COMPUTER-ASSISTED INSTRUCTION

SESSION 12B
ROOM/SALLE 143

By/Par: Glenn F. Cartwright,
University of Alberta.
Université de l'Alberta.

A central principle of computer-assisted instruction (CAI) is that students are taught individually. Seldom has the teaching of groups of students by computer been considered. In this study, 282 McGill University students, beginning a summer course in educational psychology, were selected from a group of 320 for their relatively minimal knowledge of general psychology. Subjects were randomly assigned to one of four treatments: a control group worked with CAI individually, and three experimental groups worked with CAI in pairs, or in groups of three or four. Three CAI lessons in general psychology were presented, each a week apart, by the McGill IBM 370/155 computer on eight teletype terminals. Students working alone responded individually; students working in pairs or in groups discussed the lesson material, decided on a single response, and typed it in. During the fourth week, each student responded individually to a thirty item multiple-choice test based on the three CAI lessons. Analysis of variance procedures indicated that there were no significant differences in means among the four treatments. This suggests that students learn equally well from CAI whether taught individually or in groups. Not only does group work at CAI terminals provide a dimension of human interaction not possible with individual CAI, but operating cost reductions of up to 75% are possible, depending on the group size. Further research is underway to determine the effect of social, attitudinal, and personality variables on learning with CAI in a group setting.

L'UTILISATION DE GROUPES DANS L'ENSEIGNEMENT À L'AIDE D'ORDINATEURS

Un principe important de l'instruction à l'aide d'ordinateurs (CAI= "computer-assisted instruction") réside dans le fait que les élèves ont un enseignement individuel. Il est rare que l'on ait pensé à enseigner à l'aide d'ordinateurs à des groupes. Dans cette étude, 282 étudiants de l'Université McGill furent choisis, au commencement d'un cours d'été en psychologie pédagogique, dans un groupe de 320 pour leurs connaissances relativement minimales de la psychologie générale. Des sujets ont été affectés aléatoirement à l'une de quatre méthodes: un groupe de contrôle a travaillé individuellement avec la CAI et dans les trois groupes expérimentaux les étudiants travaillaient en sous-groupes de deux, trois ou quatre. Se servant de l'ordinateur McGill IBM 370/155 à huit terminaux à télétypes, on a présenté trois leçons CAI de psychologie générale, à une semaine d'intervalle. Les étudiants travaillant seuls répondaient individuellement, ceux travaillant en groupe discutaient entre eux avant de décider de la réponse et de l'entrer dans l'ordinateur. Pendant la quatrième semaine, chaque étudiant répondait individuellement à un test à choix multiples de trente questions basé sur les trois leçons CAI. L'examen des résultats a montré qu'il n'existe pas de différences importantes entre les quatre méthodes ce qui nous amène à penser que les étudiants apprennent aussi bien en CAI, que l'instruction soit individuelle ou en groupes. Le travail en groupe présente non seulement l'avantage de relations humaines mais permet aussi de réduire le coût du fonctionnement jusqu'à 75% selon l'importance du groupe. Actuellement, nous poursuivons les recherches pour déterminer l'influence de variables sociales, de l'état d'esprit et de la personalité des étudiants travaillant en groupe.

The Use of Groups in Computer-Assisted Instruction

Glenn F. Cartwright
Department of Educational Psychology
The University of Alberta

ABSTRACT

The study traces some of the historical antecedents of computer-assisted instruction and considers the group facilitation of individual learning using CAI. During a summer course, 282 educational psychology students, screened for their lack of background in general psychology, were randomly assigned to one of four treatments. A control group worked with CAI individually, and three experimental groups worked with CAI in pairs, or in groups of three or four. Three CAI lessons in general psychology were presented. During the fourth session, each student responded individually to a thirty item multiple-choice test based on the lessons. No significant differences in mean learning scores for individuals were found among the four groups suggesting that students learn equally well from CAI whether taught individually or in groups. The study has implications for those who would reduce the overall cost of CAI by up to 75% and at the same time lessen the threat of dehumanization posed by the technology. Further research is underway to determine the effect of certain personality, social, and attitudinal variables on learning in groups with CAI.

Introduction

The development of computer-assisted instruction (CAI) has traditionally rested heavily upon the advances made in programmed instruction (PI) and teaching machines. Indeed the first reported attempt at CAI (Rath, Anderson, and Brainerd, 1959) was described as an attempt to mimic the teaching machine using a digital computer.

One of the most significant claims made in the early days of the teaching machines, was that they provided individual instruction. In truth, a single machine could do little else, and little else was expected of it. The technology had not arrived at the point where a single machine could be used successfully to replace the teacher in instructing a large class. In fact, to allay the teachers' suspicions that they might one day fall victim to automation, the claim was made that far from replacing teachers in the classrooms of the nation, teaching machines would "free" the teacher for other more creative tasks.

Just what these "creative" tasks were, was never usually specified, and whether or not teachers using PI have in fact been freed from classroom drudgery remains to be verified. Nevertheless, the potential for "freeing" the teacher helped foster an attitude of acceptance towards mechanized teaching, at least among some members of the teaching profession and an even larger segment of the public.

Since those who espoused the cause of PI and teaching machines valued the one-to-one tutorial method of instruction above all else, the fact that the teaching machine would teach only one person at a time was viewed as an asset rather than a liability. Here was the teaching machine's strength. Produced in large quantities, it could claim at least "individual" if not "individualized" instruction.

The development of computer time-sharing systems enabled CAI to follow along the well-beaten path of "individual" instruction, by providing now, not one machine, but one computer terminal per student. However, the proliferation of computer terminals, like the mass production of teaching machines, seems to have contributed only to the expansion of "individual" instruction. The exciting potential of "individualized" instruction remains yet to be realized.

It was not long however, before those in PI realized that an optimally perfected program might be able to instruct more than one student simultaneously. Several attempts have been reported (Frye, 1963; Farber, 1965; Hartley, 1966; Sawiris, 1966; Hartley and Cook, 1967; Moore, 1967; Hartley, 1968; Kay, Dodd and Sime, 1968; Frandsen, 1969; Amaria, Biran and Leith, 1969; Amaria and Leith, 1969; Hartley and Hogarth, 1971). The results of these experiments seem to indicate that individuals in pairs or groups using PI attain at least as much as do individuals working alone with PI. The learning of individuals in groups does not seem to be impaired as a result of the group experience.

These same findings hold for the few studies which have been done with pairs of students in CAI (Grubb, 1965; Goodman, 1968; Love, 1969).

With the present high costs of CAI, tremendous savings might be effected if it were found that CAI could teach groups of students as well as it could teach individuals working alone.

Two arguments stand in the way of implementation of group CAI study. The first is the belief that group instruction removes the advantage of individual pacing, long believed to be a necessary feature of individual instruction. There is mounting evidence however, that individual pacing often is not as effective as group pacing. A student's own pace is not necessarily his best pace; often students who choose a fast pace make more errors, while those who choose a slower rate frequently become discouraged with their slow progress (Gropper and Kress, 1965; Moore, 1967). It may very well be that under group-paced conditions many students would excell.

The second obstacle is the lack of distinction and therefore confusion, between the terms "individual" and "individualized" instruction. It is suggested here that the two are not the same. Having an individual working alone at a terminal does not guarantee that his needs have been met. Neither does meeting those needs require that he work alone. Truly "individualized" instruction, where the program is geared to the intellectual, social, and personality characteristics of the learner, may just as well take place in a judiciously organized group as in the solitary confinement of a carrel.

The present study was concerned with ascertaining whether or not CAI could teach individuals in groups as effectively as it could teach individuals working alone.

Subjects

Subjects in the study were 282 summer school education students at McGill University. The students were screened from a larger group of 320 students, and were chosen for their lack of background knowledge in psychology. Most of the students held at least a bachelor's degree, and were proceeding to teacher certification through three consecutive summers of a teacher education program. All were in their first summer (1971) of the three year cycle. They ranged in age from 20 to 59 years with the average age being in the late twenties.

Procedure

Each of the 282 students was randomly assigned to groups of varying sizes. Some worked alone, others in pairs, still others in groups of three or four. Each group took three CAI lessons in basic psychology, each one week apart. The three lessons were based on chapter four and five of Hebb (1966) and were originally designed for use in other studies (Roid, 1971). Each of the lessons consisted of about 35 frames, requiring the student to respond by typing in a word or phrase. Provisions made for branching to remedial sequences. Together the three lessons formed an integrated unit of work based on the nervous system, sensory control, motivation and arousal.

Students worked in their assigned groups, read and discussed the material as it was displayed, and voted on a single answer. One person from each group was designated to respond on the teletype by typing in the group answer.

The lessons were coded in MULE (McGill University Language of Education) and were presented by the McGill 370/155 RAX time-sharing system on eight teletype terminals.

One week after the final lesson, each student individually completed a thirty item multiple-choice test based on the three lessons.

Results

Means and standard deviations of learning scores for the four treatments were found in TABLE 1 below.

TABLE 1

INDIVIDUAL LEARNING SCORE MEANS AND STANDARD DEVIATIONS FOR THE FOUR TREATMENT GROUPS

Treatment Group	N	Mean	s.d.	Variance
1. individuals	78	16.22	4.49	20.15
2. pairs	61	15.89	3.83	14.64
3. groups of three	80	16.04	3.81	14.52
4. groups of four	63	15.92	4.27	18.27
Total	282	16.03	4.09	16.72

The uneven numbers in some of the treatments in TABLE 1 are due to several people working in groups during the three lessons but not completing the criterion test due to absence. Results for their partners are included since the groups were intact during the lesson phase of the project.

A simple one-way analysis of variance procedure was used to compare the mean learning scores of the four treatment groups. The results are found in TABLE 2 below.

TABLE 2

ANALYSIS OF VARIANCE RESULTS FOR THE FOUR TREATMENT GROUPS ON LEARNING

Source	df	Sum of Squares	Mean Square	F	p
Groups	3	4.813	1.60	0.09	0.96 n.s.
Error	278	4709.000	16.94		

Homogenity of Variance test: Chi square = 2.90; p = 0.41 N = 282

Discussion

It would appear from the non-significant mean differences among treatments that individuals learn equally well from CAI whether they work alone, in pairs, or in groups of three or four.

Often, studies comparing CAI with other instructional modes find no mean differences between methods, but a significant increase in the variance of learning scores for CAI. This is usually attributed to the fact that CAI is more highly individualized than other teaching methods, and the extent of individualization is reflected in the variance of the scores. Comparing the variances for individual and group CAI methods, the results in TABLE 2 indicate no significant differences. One interpretation might be that the learning of individuals in groups was just as highly individualized as the learning of those who worked alone. Other factors such as heterogeneity of the group may well influence learning score variances, but group size *per se* appears to have little effect.

Since neither group means nor variances differed significantly among treatments, it is suggested that the group use of CAI may be implemented as a legitimate (and less expensive) alternative to individual CAI, without affecting learning performance.

With respect to pacing, it may be argued that students who worked alone were individually paced, and those who worked with others were group paced. The lack of significant differences in mean learning scores between individually paced and group paced students tends to lend support to the claim that individual pacing is not a vital or even necessary feature of CAI.

From a humanistic viewpoint, the group use of the computer may help to make CAI more acceptable to those who see the technology of individual learning as dehumanizing. Elsewhere it has been suggested that the computer can act as a catalyst for social interaction (Cartwright, 1971). This removes from CAI the responsibility of providing all the interaction necessary for human learning. From the learner's point of view, having others present at the terminal may prove to be less frustrating in the case of terminal malfunction or difficulty of operation.

Assuming the same execution costs for groups as for individuals, cost savings of up to 75% may be realized with four people per terminal. With the group method, the cost of a single lesson is divided among four students. Slight variations may occur here however, since it is not known if groups take more time exploring the various branches in a lesson, or increase terminal connection time by overly long discussion of the material. Further research is underway at the moment to determine if the average execution time is longer for groups than for individuals. It should also be pointed out that the optimum group size around a terminal is not necessarily four and that larger groups may be possible, reducing costs ever further.

Further research is underway to determine if students with particular personality, social, or attitudinal characteristics benefit more from the group CAI than the individual CAI approach. In this way it may be possible to optimize learning performance by assigning students to the most beneficial CAI instructional method.

Acknowledgement

The author gratefully acknowledges the assistance of Dr. Gale Roid of the Centre for Learning and Development for providing the programs, and Prof. D. Thorpe, Director of the McGill Computing Centre, for his help with the project.

References

Amaria, Roda P., Biran, L.A., and Leith, G.O.M. Individual versus Co-operative Learning I: Influence of Intelligence and Sex. Educational Research 1969. 11:95-103.

Amaria, Roda P. and Leith, G.O.M. Individual versus Co-operative Learning II: The influence of Personality. Educational Research 1969 11: 193-199.

Cartwright, Glenn F. Social Factors in Computer Assisted Testing. Paper presented at the McGill Conference on University Teaching and Learning. Montreal: McGill University. October 20-23, 1971.

Farber, I.J. The Study of the Use of Programmed Instruction in a Group Situation. *Dissertation Abstracts* 1965. XXVI 3: 1509-1510.

Frandsen, A. Group Facilitation of Individual Learning. *Psychology in the Schools*. 1969 VI: 292-297.

Frye, Charles H. Group versus Indidual Pacing in Programmed Instruction. *AV Communication Review* 1963. 11: 124-129.

Goodman, H.J. The Effects of Team Learning and the counteracting of Misinformation upon Attitudes towards Computer Instructed Learning. Abstract of unpublished doctoral dissertation. University of California, Los Angeles, 1968.

Gropper, G.L. and Kress, G.C. Individualizing Instruction through Pacing Procedures. *AV Communication Review* 1965. Vol. 13, No. 2.

Grubb, R. The Effects of Paired Student Interaction. Paper read at the National Convention of the National Society for Programmed Instruction. Philadelphia, May 1965.

Hartley, J. Social Factors in Programmed Instruction. A Review. *Programmed Learning* 1966. 3: 3-16.

Hartley, J. Some Factors Affecting Student Performance in Programmed Learning. *Programmed Learning and Educational Technology* 1968. 5: 206-218.

Hartley, J. and Cook, A. Programmed Learning in Pairs: The Results of Miniature Experiments. *Programmed Learning and Educational Technology* 1967. 4: 168-178.

Hartley, J. and Hogarth, F.W. Programmed Learning in Pairs. *Educational Research* 1971. 13: 130-134.

Hebb, D.O. *A Textbook of Psychology*. Philadelphia: W.B. Saunders, 1966.

Kay, H., Dodd, B., and Sime, M. *Teaching Machines and Programmed Instruction*. Baltimore Md. Penguin Books, 1968.

Love, W. Individual Versus Paired Learning of an Abstract Algebra presented by computer-assisted instruction. Technical Report No. 5. The Florida State University, 1969.

Moore, D.L. Group Teaching by Programmed Instruction. Programmed Learning and Educational Technology 1967. 4: 37-46.

Rath, G., Anderson, N., and Brainerd, R.C. The IBM Research Center Teaching Machine Project. In Galanter E.(ed) Automatic Teaching: The State of the Art. New York: John Wiley and Sons, 1959.

Roid, G. Covariates of Learning in CAI in Introductory Psychology. Paper read at the AERA conference, New York City, February 1971.

Sawiris, M.Y. Group Learning Using a Projected Programme. Visual Education. August/September 1966. p. 29-31.

VOCATIONAL COUNSELLING BY MINI-COMPUTER

H.J. Hallworth, J. Gallo,
A. Herman, L. West

VOCATIONAL COUNSELLING BY MINI-COMPUTER

SESSION 13A
ROOM/SALLE 148

By/Par: H.J. Hallworth, J. Gallo,
A. Herman, L. West
University of Calgary

Several computer-based vocational and educational counselling systems have been developed which are intended to undertake some of the functions ordinarily performed by a human counsellor. The purpose is to provide the counsellor with a tool, and to offer the student a more readily available service.

Almost invariably, however, such systems have used a large time-shared computer. It was therefore considered appropriate to develop a vocational counselling system on a small time-shared machine such as a large school or small school board could afford to rent or purchase. A machine of this type could, of course, be used for a variety of other purposes simultaneously with the operation of a counselling system.

A vocational counselling system was implemented on one of the smallest time-shared computers at present available. Designed in accordance with current counselling procedures, the system is in no sense prescriptive. It assists a student in defining his interests and aspirations, and helps him to explore a variety of occupational choices. It offers some of the guidance ordinarily given by the counsellor, and may be followed up by the use of more detailed printed materials to which the student is referred.

Preliminary tests of the system were considered successful, and more extensive experiments are in preparation.

L'ORIENTATION PROFESSIONNELLE À L'AIDE DE MINI-ORDINATEURS

Plusieurs systèmes pour conseiller en matière d'éducation et d'orientation professionnelle, en se servant d'ordinateurs, ont été développés en vue d'assurer certaines fonctions qui sont habituellement celles des conseillers humains. L'objectif est de fournir un outil au conseiller humain et d'offrir à l'étudiant un service plus accessible.

Cependant, de tels systèmes ont presque toujours utilisé un grand ordinateur à temps partagé. On a, donc, pensé qu'il conviendrait de développer un système d'orientation professionnelle à l'aide d'une petite machine à temps partagé de sorte qu'une grande école ou un petit conseil scolaire puisse la louer ou l'acheter. Une machine de ce type pourrait, naturellement, servir à d'autres usages simultanément avec la mise en oeuvre d'un système d'orientation.

On a établi un système d'orientation professionnelle basé sur l'un des plus petits ordinateurs à temps partagé disponibles actuellement. Conçu en conformité avec les méthodes actuelles de conseil, le système n'est nullement rigide ou impératif. Il aide l'étudiant à déterminer ses intérêts et ses aspirations et il l'aide à choisir parmi de nombreuses occupations ou professions. Il donne certains conseils normalement fournis par le conseiller humain et il peut compléter ses conseils par des renseignements imprimés plus détaillés qui sont à la disposition de l'étudiant.

Des essais préliminaires ont réussi et des expériences plus étendues sont en cours de préparation.

There is general agreement among counsellors that the amount of information they need to have at their command is increasing at such a rate that it may soon be overwhelming. Further, the amount of time a counsellor has to spend in dispensing such information to students is such that it is impossible for him to do it adequately, and also continue to perform the more characteristically human functions of a counsellor. If we also consider the shortage of competent professionals in the field, it becomes obvious that there is an urgent need to provide counsellors with technological aids so that they may use their own time more efficiently and more economically.

The assumption underlying this paper is that it is now both technically possible and economically feasible to provide the counsellor with access to a time-shared mini-computer. He may use such a computer as a sophisticated tool, which simulates some of the simpler parts of his own counselling behaviour, as in the case of some aspects of vocational and educational counselling.

A number of computer-based vocational and educational counselling systems have been developed in recent years. The first of these was produced by Cogswell and his colleagues at System Development Corporation (1965). Others have been described by Loughary (1966), Tondow and Betts (1967), Tiedeman (1968), Impellitteri (1968), Super (1970), and Romaniuk and Maguire (1970).

In some cases student information is presented to the computer on punched cards, and the system replies by means of a printed output. For the most part, however, time-shared systems are used. The terminal equipment frequently consists of a teletype device only. Sometimes it also includes a slide projector and a tape recorder under computer control, and occasionally a cathode ray tube terminal is used. Most terminal equipment may be placed at the end of a phone line, and may be remote from the computer.

All the counselling systems referred to above, however, have in common the fact that they are implemented on a large computer. In the Computer Applications Unit of the Faculty of Education at the University of Calgary, a vocational counselling system has been developed and implemented on one of the smallest time-shared computers currently available. It was considered necessary to develop a counselling system on an inexpensive computer because, in many circumstances where such a system is needed, there is at present insufficient capital to obtain a large machine.

Further, communications costs are too high to allow operational use in a school of a terminal attached to a computer which is beyond the local phone area.

The machine on which the Calgary system is used costs no more than equipment which is routinely installed in small colleges and large high schools. The computer has a central processing unit with a 1.6 microsecond cycle time, 16,000 words of core store, one random access disk having 250,000 words, and six DECtape drives. It supports a time-shared system for 16 terminals used simultaneously. Eight terminals are on computer assisted instruction desks, each having a rear view screen and a slide projector under computer control.

At the present time the vocational counselling system uses both a teletype and a slide projector, and therefore operates at the computer assisted instruction desks. It is written in SCROLL, a language developed in the Computer Applications Unit, and requires the use of two tape drives.

From one point of view, the system is a simulation of part of the behaviour of a vocational counsellor. It is intended to perform functions similar to those ordinarily performed by a counsellor, and must therefore conform to desirable counselling procedures. However, the aim is not to replace the counsellor; it is simply to provide him with a new tool which he may use in order to employ his own time to greater effect. It is therefore necessary that computer terminal time be less expensive that counsellor time.

From the student's viewpoint, it is important that the system should provide all that the counsellor would ordinarily provide in the early stages of vocational counselling. It should, however, be more readily accessible than the counsellor. This may, indeed, be claimed as one of the principal advantages of a computer-based system.

A further consideration is that any vocational counselling system should be so designed that it readily allows revision of its data base. Conditions of employment, wages and salaries, entry requirements, etc., are changing rapidly. Information of this kind forms the data base of a vocational counselling system and has to be constantly updated.

In present counselling practice no attempt is made to direct the student into a particular type of vocation, still less into a specific occupation. He is encouraged to discover his own interests, to set his own educational goals, to explore a number of specific occupations. The object is to promote the decision-making ability of the student by making information available to him, and by giving him opportunity for vicarious experience in the making of occupational choices.

The first task in the development of the system was to devise a classification of interest areas. The categories had to be defined in plain language so that they could be understood by all high school student users and, insofar as possible, the definitions had to be such as to exclude the possibility of overlapping categories. Eleven such categories were used.

The definition of categories of educational and training aspirations presented less difficulty. High school programs are classified as matriculation, business, vocational or general. Beyond high school level, a student may follow a well defined route in terms of taking an apprenticeship or attending a technical institute, a college, or a university. Seven aspiration categories were used, ranging from "less than high school diploma" to "university".

It was necessary to have a data base of vocational materials with clear local relevance. It was therefore decided to make use of data provided by the Career Information Services of the Guidance Department of the Calgary School Board. Approximately 209 job titles so obtained were classified into the interest categories listed above. Each job description was then examined to determine the minimum educational or training level required, and on this basis the job title was allocated to one of the education and training levels. Lists were then made of all job titles within each interest/educational category. In all there were 77 interest/educational categories, with a maximum of 51 job titles in any one category. Several categories were, of course, empty.

Data for the development of the student file were derived largely from information contained in the student's cumulative folder. Each record contained an I.D. number, age, sex, high school program enrolled in, courses presently being taken, courses taken in the past and marks obtained, and total credits obtained to date. Test information such as intelligence scores, interest inventory scores, and aptitude test scores were also included.

When brought into use, the system immediately informs the student that he may at any time type QUIT to terminate the session, or RETURN to go back to the start. The two modes of operation are then explained and he is asked to choose one of them. If he chooses the "exploration mode" he is presented with a slide showing the list of interest categories. He determines his interest area and types in the appropriate number. A record slide is then presented, showing the list of educational and training categories. Again he makes his choice by typing in the appropriate number. A third slide is then shown, having a list of appropriate job titles. Each such slide is preceded by an appropriate code number. Upon typing in the code number of a job, he is presented with a brief description of the job, special educational qualifications required, length of training, employment opportunities, wages, etc.

The student's record is then compared with the job requirements and a series of messages is printed at the terminal. These tell the student whether he is presently in the correct "route" in school to become qualified for this job, how many credits are required for entry to the job and at what level, how many credits he has already acquired and how many more he will need upon successful completion of his present courses, any specific subject requirements for the job, and which of these he has already met. No evaluative comments are made: the student is merely presented with the facts and is allowed to make his own decisions. He is given a reference to the appropriate career information booklet published by the Calgary Public School Board, and he may use the job code number to obtain further information from the school counsellor.

The system then asks the student whether he wishes to quit, return to the same list of job titles, return to the beginning of the exploration, or return to choose another mode.

The "index mode" simply provides, by means of slides, an index to all job titles so that they may be accessed directly. It is especially helpful to students who wish to obtain information about specific occupations. The terminal first presents a summary index, from which the user may obtain the alphabetic section of the index including the job title in which he has a special interest. Each job title is preceded by a code number and upon typing in this number the student may obtain the job summary. In each case he also receives a printout of the comparison between his own record and the job requirements, as already described. At any point he may quit, return to choose another mode of operation, or return to the summary index in order to choose another alphabatic section.

The counsellor may use the exploration and index modes. He also has a third mode, in which he obtains direct access to the student data file.

The system has undergone preliminary testing. Thirty six high school students were drawn in equal proportions from males and females in Grades 10, 11 and 12. Each used a terminal for four sessions of 25 minutes. Using as a criterion the number of jobs examined, there was no evidence that use of the system was related to either language ability or clerical speed and accuracy. There was a significant correlation between intelligence and the number of job examined. However, analysis of covariance indicated no differences in the amount of use by grade or by sex. There was no reason to assume that the system was not equally usable by students of both sexes and at each of the three grade levels.

A Likert-type questionnaire and a brief essay both indicated very positive attitudes towards use of the system. Students found the teletype easy to use, they wanted more time on the system, and they suggested extension of its use to Grade 9.

The system has been very well received by counsellors. It is therefore now being modified and prepared for more extensive testing.

In effect, experience to date suggests that further experimentation is warranted, and that a useful and economically viable system can be developed.

ACKNOWLEDGMENTS

The project reported in this paper has been supported in part by the Alberta Human Resources Research Council.

The occupational data used in the system were extracted from Career Trends, with the permission of the Calgary Public School Board.

REFERENCES

Cogswell, J.F., and D. P. Estavan, "Explorations in Computer-Assisted Counselling," System Development Corporation Document, TM-2582/000/00. August 6, 1966

Impellitteri, J. T., "The Development and Evaluation of a Pilot Computer-Assisted Occupational Guidance Program," Final Report to the Pennsylvania Department of Public Instruction, Project No. 16033, 17023, 18033, University Park, Pennsylvania, Department of Vocational Education, The Pennsylvania State University, July, 1968

Loughary, J. W., Friesen, D., and R. Hurst, "Autocoun: A Computer-Based Automated Counseling Simulation System," Personnel and Guidance Journal, September, 1966, pp. 6-15

Romaniuk, E.W. and Maguire, T.O., "Computer Assisted Guidance," Canadian Counsellor, June, 1970, Vol. 4, No. 3, pp. 149-160

Super, D. E., "Using Computers in Guidance: An Experiment in a Secondary School," Canadian Counsellor, Vol. 4, No. 1, January, 1970

Tiedeman, D. V., et al., "Information System for Vocational Decision-Making," Second Annual Report, Cambridge, Mass., Graduate School of Education, Harvard University, 1968

Tondow, M., and M. L. Betts, "Computer-Based Course Selection and Counseling," Journal of Educational Data Processing, 1967, pp. 216-241

THE ECONOMICS OF VARIOUS SCHEMES TO AUTOMATE FRENCH TEACHING

Raymond Lamérand
Language Bureau
Centre linguistique d'Ottawa
Chemin Carson 800
OTTAWA ONTARIO K1A 0M7

THE ECONOMICS OF VARIOUS SCHEMES TO AUTOMATE FRENCH TEACHING

SESSION 13B
ROOM/SALLE 148

By/Par: Dr. R. Lamérand,
Centre linguistique d'Ottawa,

Not all aspects of language-teaching can be automated; some facets of language-learning seem adequately catered for by conventional programmed instruction techniques and automation appears superfluous; many parts of a French course can best and most cheaply be mastered with the help of computers.

The parameters to be considered are the time the student requires to reach criterion performances in listening comprehension, memorisation and adaptation of conversational formulas, a measure of self-expression using grammatically correct constructions and appropriate morphemes, and reading comprehension at various levels of complexity; the efficiency of instruction aimed at helping the student reach the goals envisaged; the cost of elements in the system financed by the teaching authority - administrative, hardware and software.

Automation of some aspects of administration (including the teacher as administrator in the classroom) leads to computer-managed-instruction with certain fixed costs being compensated by a number of educational advantages; automation of certain skills allows the teacher to concentrate on creative aspects of language use in a group situation - and the lack of this element in many teaching contexts leads to a high rate of failure in meeting objectives of foreign-language courses. Automation alone can provide certain information to improve the efficiency of phonetic differenciation as an element in language teaching.

LES ASPECTS ÉCONOMIQUES DE DIFFÉRENTES MÉTHODES DE L'ENSEIGNEMENT DU FRANÇAIS À L'AIDE D'ORDINATEURS

Il n'est pas possible d'adapter l'enseignement à l'aide d'ordinateurs à tous les aspects de l'enseignement d'une langue; certains semblent se prêter d'une manière satisfaisante à des techniques d'enseignement programmé conventionnelles et l'automation parait superflue; l'ordinateur, en ce qui concerne beaucoup d'éléments d'un cours de français, s'avère le moyen le plus efficace et le meilleur marché pour l'apprendre.

Les paramètres dont il faut tenir compte sont: le temps nécessaire à l'étudiant pour atteindre le niveau exigé en écoutant, en comprenant, en apprenant par coeur et en adaptant les formules de conversation, en pouvant s'exprimer en termes grammaticaux corrects avec les morphèmes appropriés et en comprenant la lecture à divers niveaux de difficulté; l'efficacité de l'enseignement pour que l'étudiant arrive aux buts proposés; le coût des éléments du système financés par les autorités de l'enseignement, c'est-à-dire l'administration, les ordinateurs et la programmation.

L'automation de certains aspects de l'administration (y compris le professeur en tant qu'administrateur dans la classe) mène à l'enseignement géré par ordinateur où certaines dépenses fixes sont compensées par des avantages éducatifs; l'automation de certaines techniques permet au professeur de porter toute son attention sur les aspects créatifs de l'emploi de la langue dans un groupe et, dans beaucoup de milieux enseignants, on n'atteint pas les objectifs des cours de langues étrangères en raison du manque de cet élément. L'automation seule peut fournir certains renseignements qui serviront à améliorer l'efficacité de la différenciation phonétique dans l'enseignement d'une langue.

Automated French Teaching requires appropriate hardware, a satisfactory programming language and a suitable team for the physical production of the pedagogical materials. For a scheme to be economically viable, capital investment and recurring maintenance and communication costs over a realistically short period must be distributed over a large enough population to be competitive with cheap non-automated systems.

The cost of teaching French in any area is intimately bound up with the size of the population to be reached and its geographical distribution. There are areas where students of French are so few in certain levels of competence that conventional teaching can become uneconomic. It solves nothing to regroup arbitrarily learners in the same field regardless of disparity of background and achievement. Individualized methods of instruction may free a teacher so that he can deal with his students on a one-to-one basis; good programmed courses are marketed to meet a variety of needs. However, no individualized courses and no computer-based systems can render superfluous contact hours between a human teacher and the learner.

No matter where a student lives, it is essential to budget for adequate contact between the instructor and his charges. The human teacher is indispensable for a variety of activities involved in language acquisition. Depending on the sophistication of a particular curriculum and the purposes for which the language is being learnt, the teacher is needed

to solve problems which the individual learner does not surmount unaided
to correct faults of pronunciation if desirable
to render the student critical of his own efforts
to encourage the student to use language creatively (free self-expression either orally or in writing)
to guide the student in making constructive use of his time
to orient the learner to derive the greatest possible benefit from the pedagogical materials at his disposal
to motivate the trainee as required...

The instructor complements the programme and fills in for such deficiencies as may occur. Thus, where the student can type in comments while working on the lesson materials, the teacher must be available to provide satisfactory feed-back. Where automated lessons do not provide sufficient remedial texts and do not arrange class room sessions, the administrative work involved falls on the teacher. As a rough estimate, some 25% of existing costs for traditionally administered instruction will still need to be provided to complement any scheme to automate language teaching. Provision must also be made in the budget for such retraining of staff as may be occasioned by the transition to computer-aided learning. This is not an area where the onus should be put on the teacher to seek further in-service training.

The objectives of a particular French course determine the teaching priorities and influence the choice of computer equipment and language facilities. If adults are to be taught no more than comprehension of the French periodicals and technical correspondence that come to their desks, there is no need to install costly equipment to handle speech input and output. When the teaching system provides such facilities for other reasons, it could become advantageous to play pre-recorded messages back to the student for pedagogical reasons if the added production costs are not disproportionately high. Necessity and cost-effectiveness are prime considerations in evaluating what ingredients should be made part of any particular computer-based educational installation.

Again, no matter what the cheapest proposition may be in terms of efficiency and per capita expenditure, some school districts may simply be unable at any given time to make available adequate funds to meet the initial outlay. Optimum solutions may remain illusory until cheaper communications and willingness to learn from others make inter-district and even inter-provincial boundaries irrelevant. Rivalry and ambition could easily render meaningless any plans in the area of computer-aided learning.

Given existing Canadian telephone company policies, it could well be that automated instruction in low-density population areas would be cheapest if conducted in terms of a small computer serving a hundred or so terminals. The MITRE Corporation claims that its TICCET system is unique in being able to handle up to 128 television screens at one moment, on the basis of "updating" (not refreshing) sixty student stations each second [1]. While programming in a multi-purpose language like PL/1 can be time-consuming and hence expensive, the costs of developing an educational language would immediately increase the expense of the TICCET package. However, when teachers experience too much difficulty or tedium in preparing lessons for the computer, they will allow the system to remain unused. In

[1] K.J.Stetten and others, *The Design and Testing of a Cost Effective Computer System for the CAI/CMI Application*, MITRE Corporation, January 1970, M69-39 Rev. 1, pp. III and 10.

such cases, any use the equipment happens to get could become prohibitively expensive in terms of cost per hour per student just as installations that can be used for evening classes - beyond the eight-hour day of peak activity - tend to become quite cheap on a per capita basis.

Hardware essentials for language teaching are hard to define. How effective language instruction without a computer controlled sound source can be is evidenced by the German teaching experiment at the State University of New York at Stony Brook. The control group followed an audio-lingual course with some sixty minutes per week in the language laboratory; in the latter part of the first year students devoted more time to reading and writing. The experimental group had no language laboratory sessions, but worked from the first on reading and writing skills with the help of a computer. Like the others, their classroom work followed the audio-lingual pattern. All students had similar study assignments and work-loads. The fact that CAI students (85%) performed significantly better in reading and writing than the median student in the non-CAI population is attributed to the greater efficiency of CAI teaching. Experience with the PLATO III system at the University of Illinois was similar for the teaching of French. CAI students rapidly became competent in translation skills, including phonetic transcriptions for French majors. Teachers compensated for what could not be done at the computer terminal.

CMI strategies could involve instructing a student to listen to pre-recorded tapes and discs or to perform given language laboratory exercises. The computer terminal need not necessarily have equipment installed to present such materials. P. Rosenbaum and S. Issakow produced 126 hours of a Russian course where the computer controlled a sound source, but their objective was to automate the acquisition of listening comprehension and dictation skills. It was therefore possible, though perhaps not imperative, to have aurally cued grammar exercises. Work with troops at Monterey does not prove conclusively that computer control of the sound source was particularly beneficial for learning dictation skills. What students learnt, they assimilated thoroughly and retained well with impressive rapidity. What can be questioned is the desirability of using existing computer techniques to have students type Russian when they will later be required solely

1. P. Rosenbaum, The DLI-IBM Feasibility Study in Computer-Assisted Foreign Language Instruction (1969), IBM Report 2595 publication elsewhere could not be found.

to write freehand in that language. The type of equipment the computer controls at the student console could be best determined from the achievement objectives set for a course.

The amount and nature of student terminals decide the minimum desirable size for the central processing unit of the computer at the heart of a CAI system. Obsolete computers if bought cheaply can sometimes appear adequate for satisfactory teaching. One must be careful, however, that desired peripherals are not incompatible with the computer or that unacceptably long delays due to the reduced size of the machine make for unsound pedagogy. Usually the student needs the illusion that he is the sole user of the learning facilities. It is fortunate that IBM policy on rentals of obsolescent equipment is such that it is cheaper to implement COURSEWRITER III on their newer systems than to look for "cheap" alternatives.

A major factor in the cost of CAI is the size, qualifications and programming time required by the writing staff. When off-the-shelf programmes are not available to meet the language teaching objectives of a particular school or business, it is imperative to evaluate how expensive per student hour a CAI course is going to be. In the case of the DLI Russian course referred to above, professional staff contributed 1271 hours to initial authoring and 338 to revising materials for an average 12.7 hours for each of 126 course hours. Such a low figure goes contrary to usual estimates for COURSEWRITER authors who are expected to take eight to fifteen times more hours to prepare the finished programme. Adjunctive programming as a style is faster to achieve than other types and the guidance of Issakow undoubtedly speeded the work of authors. It would in most cases be unrealistic to imagine that a teacher working alone in producing COURSEWRITER III materials could match such an achievement. Feeding partially written course material to a professional programmer [1] or to a type of precompiler like VAULT at the University of Alberta could diminish expense considerably [2]. Such a proceeding relieves CAI lesson writing of much of its tedium and repetitiveness. The use of "macro" instructions has a similar effect on authorship [3].

1. M.W.Dowsey, *Towards a True Author Entry System for CAI*, *Programmed Learning and Educational Technology* 7 (1970) pp. 43-55 tries to resolve the dilemma of the author who does not want to learn how to code his lesson for the computer. He favours the work of Dean and Grubb at San José.
2. W. Birtch and others, *VAULT Concepts and Facilities*, Edmonton, 1969, 24 pp. and *Research and Information Report VAULT*.
3. Apart from provision made in a language like COURSEWRITER III see *A Macro System for CAI*, US Army Command, 1970, 72 pp.

The TUTOR language used with the PLATO project at the University of Illinois has in fact altered traditional estimates for the time required to produce a CAI lesson. It is generally asserted that a teacher will learn the basic techniques of writing in TUTOR in some five hours or so at a student terminal. One teacher who wrote 3 teaching hours for the computer took some 26 hours to create the lesson. What a language teacher would require to write a listening comprehension lesson in French for PLATO IV remains an unknown, but it is likely to take no more than Rosenbaum and his associates took for the DLI Russian programme. The figures involved are therefore comparable to acceptable estimates for text-book writing, especially when distributed over the writing of a year's course and not limited to a demonstration lesson. The availability of specialized consultative help for certain aspects of course authoring makes for more effective and faster lesson production prior to initial presentation to students and diminishes the amount of revision needed subsequently.

In 1971, J. Brahan of the National Research Council suggested that a sample segment of the Language Bureau French programme be adapted for computer presentation using CAN. The presence of staff from the Ontario Institute for Studies in Education for two days while we became acquainted with the CAN language and wrote the main sub-routines for the presentation of instructional and test sequences greatly speeded this aspect of the course preparation. The subsequent work of producing the tape-script and the written text, of devising visuals and of having recordings made, of editing and addressing the audio-visual source materials was a team effort requiring the collaboration of many services. The adaptation of CAN for language teaching goes hand in hand with developments in available hardware and the redefinition of what aspects of language teaching can be automated at a given stage of development. The more disparate elements are to be integrated into a single lesson presentation, the longer it will take to produce the course. On the other hand, the more effective the instruction, the less will be the time taken by the student to achieve criterion levels of performance. When a CAI curriculum is designed, it is misleading to calculate costs solely in terms of the hours taken to prepare a one hour lesson for the student. Surely the only valid measure to allow comparison with traditional forms of instruction takes into account the time students require to achieve the same objectives with the help of a computer. When students reach their goals in half the time required for classroom instruction, educatinnal expenses are likely to be reduced in a variety of ways.

Even if CAI programmes written by this or that pedagogue can be proved more efficient teaching tools which make for better retention of content, more effective development of basic skills and all in a fraction of the time required in the schools of a given locality, the money spent in leasing or amortising purchased hardware is excessive with the computer configurations to which we are accustomed. Besides, their very efficacy creates problems: what are we to do with the twelve-year old who qualifies to embark on University studies? How can we cope with employment needs of twenty-year old PhDs, M.D.s and the like? Is Canadian society anxious to pay for accelerated development? The savings made in reducing the time devoted to state-subsidized formal education are liable to be lost through provincial grants to very youthful unemployed graduates. Only government and business agencies can derive immediate profit from training management to speak the second official language more efficiently and quickly.

The team at the Ontario Institute for Studies in Education who gave us the CAN family of computer languages subsidize Canadian CAI teaching. In the United States, one of the major costs most institutes engaging in CAI faced was the creation of a satisfactory tool for teaching. In Canada, users of CAN who meet programming problems can get tailor-made improvements to available facilities through cooperation. National Research Council work on hardware also ensures that taxpayers' money is used to best advantage. Experimental equipment for use with the NRC PDP-10 currently involves costs of the order of $10600 for each student terminal and a like figure for basic interface hardware permitting communication at 1200 bit/sec. These audio-visual terminals include unique features like the touch-sensitive graphic input device designed by A.M.Hlady[1]. We can be confident that current work on a speech input device will in due course make language teaching even more flexible than it can be today.

Such CAI schemes could be classified as ad-hoc frame-oriented teaching systems where the programmer specifies subject matter for the student, poses particular questions and devises strategies for coping with individual foreseen answers. J. Carbonell contrasts these approaches with information-structure-oriented teaching[2]. In this most sophisticated of inter-

1. *An X-Y Touch Sensitive Position Encoder for Computer Input*, *Bulletin of Radio and Electrical Engineering Division* 19, 1969, no 3, pp. 15-19
2. *Mixed-Initiative Man-Computer Instructional Dialogues*, Cambridge (Mass), Bolt, Beranek & Newman Inc., report 1971,1970

active systems, instructors allow the student initiative in utilizing the computer resources at his disposal and the conception of automated instruction changes. The student can consult a network of facts, concepts and procedures; at the heart of the system is what we might call a "semantic" grid which interrelates the lexis. Lexical units are type_nodes when they point to an informational, multi-level list; they are token_nodes when they refer back to one of the types. Information in the programme is stored once only without redundancy; synonyms are stored as such. Each element is categorized by specifying a set of properties: the name of the attribute, the tags necessary for its correct use and the value of the property (a set of properties or a pointer to a (set of) unit(s)). While Quillian on whose work such a system depended for its elaboration now chooses to make his whole semantic network a net of pointers [1], it is desirable for practical purposes to differenciate type and token_nodes. The SCHOLAR system provides for the teacher to input the relevant subject matter, to define the conditions for student-computer interaction and to specify what data he requires on the learner's behavior. The student working with the computer can answer the computer's questions, pose his own and receive information or be tested on his knowledge. Some hundred or so stereotyped sentences of considerable variety complement the lexicon and allow print-outs like: "You have made ... mistakes. I suggest you devote more study to ... " The implementation of SCHOLAR on a PDP-10 makes for acceptably rapid interaction between the student and the machine, but its effect on cost-effectiveness is difficult to calculate.

A similar type of facility is "dialog" within the context of the PLATO project at the University of Illinois [2]. As implemented for the large-scale PLATO-IV version, information-structure-oriented teaching is only one aspect of the TUTOR language conceived by P. Tenczar. A variety of foreign-language courses are taught with great efficacy. K. Myers, R. Ariev and others have adapted F. Marty's Active_French programme and made it the foundation of undergraduate instruction in the language. At this stage, active updating of the computer programme from the PLATO-III to the PLATO-IV version is in

---

(AD 707 782). While not the only statement of the pedagogical technique, this paper in its 211 pages is the clearest available of this type of computer-based learning.

[1] M.R.Quillian, Semantic_Memory, Pittsburgh, Carnegie Institute of Technology PhD thesis, 1966 and The_Teachable_Language_Comprehender: A_Simulation_Program_and_Theory_of_Language, Comm._of_the_A.C.M. 12, 1969, no 8, pp. 459-476.

[2] J.Riskin, Written_Composition_and_the_Computer, Urbana,

progress and F. Marty is devoting considerable effort to developing listening comprehension lessons for the course. Author costs could be met by a charge of three cents per student per terminal-hour which would guarantee the equivalent of royalties on a publication. Publishing and distribution costs met in printing houses are here incurred by the Institute providing the CAI lessons. The latter "sell" themselves in the measure that they are effective tools for learning a language. What justifies the initial investment is the high prospective enrollment. In the last analysis, an Institute must budget for its hardware and software costs in the context of the allocation for teaching which also comprises contact between teachers and their students. However cheap it may be to write lessons, however effective may be the simulations and pupil-machine dialogue, however many times more efficiently certain linguistic skills may be acquired through TUTOR, the PLATO-IV system may not be economically feasible outside the sphere for which it was created. What is ideal for large-scale use in high-density population areas must be analysed anew when Canadian teaching involves sparsely populated regions where communication charges render a remote computer prohibitively expensive to use at all.

Language teaching can involve a limited number of types of exercises and tests and a limited number of contexts appropriate for different groups. Skeleton programmes can be devised which require of the teacher to specify what his particular objectives are, the age and interests of his students and the type of equipment locally available. A French programme conceived in such terms could then conceivably be implemented differently in various locations and meet the specific requirements of each user who makes his own contribution to the design of the programme adopted. It is illusory to imagine that one can quickly adapt a given text-book or even programmed course. CAI involves completely different strategies and has to be designed as such from the start. The mere use of a computer as though it were a kind of primitive teaching machine dispensing programmed instruction is doomed to remain an unjustifiably expensive game. The design of "drive" components for PLATO, "module" formulation for the DLI Russian programme or CAN sub-routines mentioned above make CAI capable of adapting lessons to the learner and justifying the expense of man-machine interaction.

University of Illinois, 1971, CERL Report X-31, pp. 4 ff.

THE TEACHER-AUTHORED INSTRUCTION MANAGER
(TAIM)

by

Marv Westrom
Graduate Student

Division of Educational Research Services
University of Alberta
Edmonton, Alberta

THE TEACHER-AUTHORED INSTRUCTION MANAGER (TAIM)

SESSION 14A
ROOM/SALLE 145

By/Par: M. Westrom,
University of Alberta.
Université de l'Alberta.

TAIM consists of a set of computer programs that are designed to provide the regular classroom teacher with the means to use a computer as an instruction-management assistant. TAIM has no built-in curriculum; rather, the teacher must supply a set of displays, tests, and logic to the computer. The computer then assembles the displays and tests according to the supplied logic to provide individualized output for the students. TAIM has facilities for scoring tests and keeping records of all transactions. It provides the teacher with advance warning of student assignments and allows the teacher to over-ride the resident logic. The teacher has the ability to modify or extend the displays, tests or logic at any time, and TAIM monitors such changes to ensure that faulty logic is not included.

Displays may consist of any printed material. This would range from short directions (perhaps to view a filmstrip or read a portion of text) to detailed explanations of a concept. The tests must be of a multiple choice or integer-value-answer type so that the computer can mark the responses, but teacher-marked test scores can also be read-in. The logic consists of six possible statements: requesting a display or test, requesting a message to the teacher (perhaps a warning that a student has entered a particular section), conditional displays or messages, and conditional or unconditional branches to other portions of the logic file.

Using the TAIM system, a classroom teacher may be expected to improve (through repeated formative evaluations) the lessons given to students and the sequence in which students receive lessons. By assuming the chores of lesson distribution, test scoring and record keeping, the TAIM System is expected to allow the classroom teacher the means to provide effective individualized instruction.

PROGRAMMES POUR AIDER LE PROFESSEUR DANS SA FONCTION DE GESTION (TAIM)

TAIM consiste en un ensemble de programmes conçus pour donner au professeur de classe le moyen d'utiliser l'ordinateur pour l'aider dans sa gestion. TAIM n'a pas de programmes fixes; c'est au professeur de fournir des affichages, des tests et une logique à l'ordinateur. L'ordinateur rassemble alors les affichages et les tests selon la logique pour donner du travail à chaque étudiant individuellement. TAIM peut noter des tests et enregistrer toutes les opérations. Il signale à l'avance, au professeur, les devoirs des étudiants et lui permet de passer outre la logique stockée. Le professeur peut, à tout moment, modifier les affichages, les tests ou la logique et TAIM vérifie ces changements pour s'assurer qu'une logique erronée ne s'y trouve pas.

Les affichages peuvent prendre la forme d'imprimés variant entre des directives brèves (passer un film ou lire un texte) et des explications détaillées d'un concept. Les tests doivent prendre la forme d'un choix multiple ou avoir des réponses, mais les notes données par le professeur peuvent être aussi affichées. La logique se compose de six énoncés possibles: demander un affichage ou un test, demander un message au professeur (par exemple, un avertissement que l'étudiant a commencé une certaine section), des affichages ou messages conditionnels et des dérivations conditionnelles ou non vers d'autres secteurs du fichier de la logique.

En utilisant le système TAIM, le professeur de classe peut s'attendre à une amélioration (par des évaluations formatives répétées) de ses leçons et de leur séquence. En répartissant les leçons, en notant les tests et en tenant les fichiers, le système TAIM doit permettre au professeur de pouvoir donner un enseignement efficace et individualisé.

THE TEACHER-AUTHORED INSTRUCTION MANAGER

The Teacher-Authored Instruction Manager (TAIM) is a set of computer programs which facilitates a Computer Managed Instruction (CMI) environment. Where most CMI systems of instruction available to date restrict and are dependent upon the nature of instruction for which they are used, TAIM may be more properly considered as a teacher tool. That is; while the communication links between teacher, computer, and student are defined by TAIM, the nature, scope, media, mode, sequence, and rate of instruction are not.

In effect, TAIM is an instructional assistant with the following capabilities:

1. storage of teacher-prepared lessons and tests,
2. storage of teacher-prepared decision algorithms (decision algorithms are the rules by which it is decided which instruction an individual student will receive next),
3. retrieval of specific lessons and tests for individual pupils according to the particular decision algorithms he or she encounters,
4. automatic scoring of multiple choice and integer-value-answer tests,
5. automatic storing of test scores and student sequences,
6. rapid retrieval of specific information on students and across students, and,
7. easy modification of lessons, tests, decision algorithms, and the sequence of individual students.

One of the main criteria in the design of TAIM was the division of necessary teaching tasks into those capable of being done by computer and those which require human attention. The majority of a teacher's duties are complex; of these, a good portion are such that algorithms can be devised for their execution. The computer is not capable of devising algorithms, nor of ensuring that they are working properly - these jobs must be done by a human. But the computer is naturally more adept and efficient at executing complex processes that have been prespecified. In an individualized setting where the teacher's time is at a premium, it is clearly useful to free him for tasks which he alone can do.

There are three broad categories of tasks that fall to a teacher using TAIM:

1. devising and revising instructional material and algorithm rules for the computer to execute,
2. monitoring the actions of the computer and the reactions of the students to ensure that instruction is proceeding effectively,
3. giving individualized personal aid to pupils when their instruction as supplied by the computer has failed, diagnosing the reason for failure, and making the appropriate revisions to the material and/or algorithms.

The following section 'The Structure of TAIM' discusses the ways in which the teacher prepares information for storage in the computer. The section 'Lesson Production' presents a description of how the computer assembles individualized lessons for students from this information, and how communication between teacher and students through the computer takes place. Under 'Online Capabilities', the teacher's power of control over the computer is made explicit and the mechanics of the teacher/computer interface explained. The final section of this paper attempts to predict the impact for education of the TAIM system.

The Structure of TAIM

TAIM keeps about ten information files on random access disc storage attached to the computer. Of these, four are central to the operation of the system: the Display file, the Test file, the Records file, and the Logic file. Each of these files is partitioned into a variable number of 'units'. Each unit is labelled so that it may be referred to by both the TAIM user and the computer programs. Units of different files may have the same label; where confusion might arise, the labels are prefixed with D- (for the Display file), T- (for the Test file), R- (for the Records file), or L- (for the Logic file).

TAIM keeps one set of these files for each grade group using the system. However, as individual students meet or do not meet the criteria set for them by the teachers, they are channeled to different portions of these files. In a sense, the Logic, Display, and Test files specify a curriculum for the course; individual students proceed at different rates and respond differently so that they access different parts of the total course specification. The Records file keeps track of their past and present positions, and their Test results.

The Display file has the simplest structure of the four. Each labelled unit is called a Display and consists of a variable number of lines of textual material. Displays are printed out for students on a high-speed printer by the Offline computer program. The contents of Displays may range from short instructions to a student (to read a portion of a textbook, view a filmstrip, meet with a resource person, etc.) to lengthy discussions of concepts. In short, Displays consist of any printed material that a teacher might wish to be given to an individual student.

The Test file contains a variable number of labelled units called Tests. Each Test has two parts: a textual part and a 'weights' part. The textual portion is similar to a Display and is the part printed out for students. This part would normally contain a number of test questions and instructions for the recording of responses to these. Since the responses must be capable of input to and analysis by the computer, the questions are restricted to either multiple choice or integer-value-answer variety. The Weights part of each Test contains coded information for the marking of student responses, and so constitutes an answer key to the Test.

The Records file contains a unit for each student registered for the course. Each unit is called a Student Record and is labelled with a student's identification such as the first five letters of his last name followed by two initials. These records contain registration information (such as the student's age, sex, grade, home room number, etc.) and a complete record of all lessons (Displays and Tests) he has taken, the marks for all Tests, and a pointer indicating which lesson he is to be given next. (It also contains NOTE and HOLD information discussed in section three of this paper.)

The Logic file contains a variable number of labelled units called Days. Each Day consists of two parts: a lesson part and an algorithm part. Each part contains coded statements which control the assembly of Displays and Tests into individual printouts or specify the decision algorithms; each coded statement consists of a keyword followed by a number of parameters.

When constructing a lesson for an individual student, TAIM proceeds by writing the student's name at the top of a page, locating the particular Day that the student is at, and then executing the coded statements sequentially. The possible coded statements for the lesson construction are:

SHOW D-label	This statement causes a printout of the contents of the Display labelled D-label from the Display file to be made.
TEST T-label	This statement causes a printout of the textual portion of the Test labeled T-label from the Test file to be made.
MESG "message"	When an individual student encounters this statement, his name, the date, and the "message" are sent to the teacher. By placing these appropriately, the teacher can cause the computer to warn him of occurrences that require his attention. Eg: a student entering a difficult section.
IF (Condition) SHOW D-label IF (Condition) MESG "message"	The IF statement is used to cause a conditional Display to be printed or message to be queued. Although many possible forms of Conditions are conceivable, the current version of TAIM is limited to comparisons of Test results and counts of the number of times a student has passed through a particular Day of the

Logic file. In a Condition, an L-label represents the number of times the particular student has been given the named Day; a T-label represents either the student's total or his percentage score on the named Test.

Conditions may be either Single Conditions or Double Conditions. The Single Condition may be either (L-label R Value) or (T-label R Value) where the L- or T- labels take the values described above, R is replaced by one of =, ¬=, >, >=, <, or <=, and Value is replaced by a non-negative integer with an optional per cent (%) sign following. The Single Condition is True if the value of the label bears the R relationship to the integer Value, False otherwise. If the student has not encountered the L-label, or has not taken the Test named by T-label, then the Condition is always False.
Examples: Suppose a student has encountered L-CHAP3.1 twice, and received 12 out of 20 marks on test T-CH3.1 - then;

(L-CHAP3.1 > 1) would be True for this student,
(T-CH3.1 < 15) would be True for this student, and
(T-CH3.1 < 15%) would be False (his %-mark is 60%).

A Double Condition consists of two Single Conditions joined by either AND or OR. If the connector is AND, then the Double Condition is False if either (or both) of the Single Conditions are False. If the connector is OR, then the Double Condition is True if either (or both) of the Single Conditions are True. The Double Condition is always False however, if either (or both) of the labels have not been encountered by the student.

Examples: (L-CHAP3.1 = 3 AND T-CH3.1 > 50%) would be False, and
(T-CH3.1 > 10 OR L-CHAP3.1 < 2) would be True.

T-labels may be used to refer to portions of a Test by placing question numbers after them. T-CH3.1(1,2,8,9,10) would take a value equal to the total of the marks for items 1,2,8,9, and 10 of Test CH3.1 or the percentage of this total over the total possible marks for these items.

There is clearly a variety of possible extensions to the proposed Conditions. However, the means of constructing Tests and inputting data make the possible T-label values quite flexible. For example, if a teacher wished to discriminate of student I.Q., he could make up a single question test, label it 'T-IQ', and then code and read-in cards with each student's actual I.Q. on them. Afterwards, any conditional statement such as IF(T-IQ <= 110)SHOW REV3.3 would be processed appropriately.

The second part of each Day in the Logic file contains the decision algorithm specifications. There are only two acceptable statements:

GOTO L-label	This statement causes the student's next-lesson pointer to be changed to L-label.

WHEN(Condition)GOTO L-label	This statement causes the student's next-lesson pointer to become L-label if the Condition is True. Condition is defined as above.

These statements are used to determine which Day a student will be assigned to when he has finished the current one. The statements are examined sequentially (for each student) until the next-lesson pointed is set by a WHEN statement with a True Condition or by a GOTO statement. Each Day must have exactly one GOTO statement which is last in the unit. There may be any number of WHEN statements.

Lesson Production:

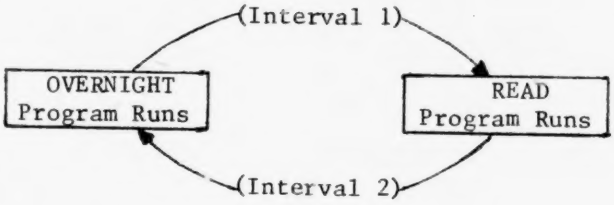
There are essentially two steps in the production of lessons for individual students under TAIM. These steps correspond to the two parts of each Day in the Logic file.

The first step occurs when TAIM consults the next-lesson pointer of an individual student to determine which Day he will receive. This Day is looked up in the Logic file and the lesson construction statements found there are executed. The student receives his lesson printout, does the indicated work, and turns in a computer card. If the lesson contained a Test, the student should have penciled his responses on this card.

Step two occurs when the card is read by TAIM. If a test was given, the student's responses are marked and the results stored in the student's Student Record; then the decision algorithm part of the Day is executed and the student's next-lesson pointer set accordingly.

Although these steps are closely related, they are executed at different times by different programs. Step one occurs during an overnight off-line program run. When this program starts, all students' next-lesson pointers are determined and available. Lessons are printed for students on the high speed printer through interpreting the necessary lesson construction statements. When finished, the program leaves all students' next-lesson pointers undetermined (that is, not pointing to a next lesson), but keeps a record of the algorithm statements to be processed to determine the pointer for each student.

The next day, the bundle of lessons is delivered to the school and distributed among the students. After doing their assigned work and preparing their computer cards, these cards are read-in by TAIM. The READ program marks all necessary Tests and then interprets each student's decision algorithm statements to set their next-lesson pointer. That evening, the overnight program runs again and the cycle repeats.



During both intervals 1 and 2, the teacher may execute other online commands of the TAIM system. These commands are outlined in the next section; they allow the teacher to gather information about student performance and the contents of students' lessons, and to alter student's next-lesson pointers. The two figures on the following page diagram the operation of the overnight and READ programs. The figure following these shows the communication links between the school and the central computer (a telephone line and a courier delivery service).

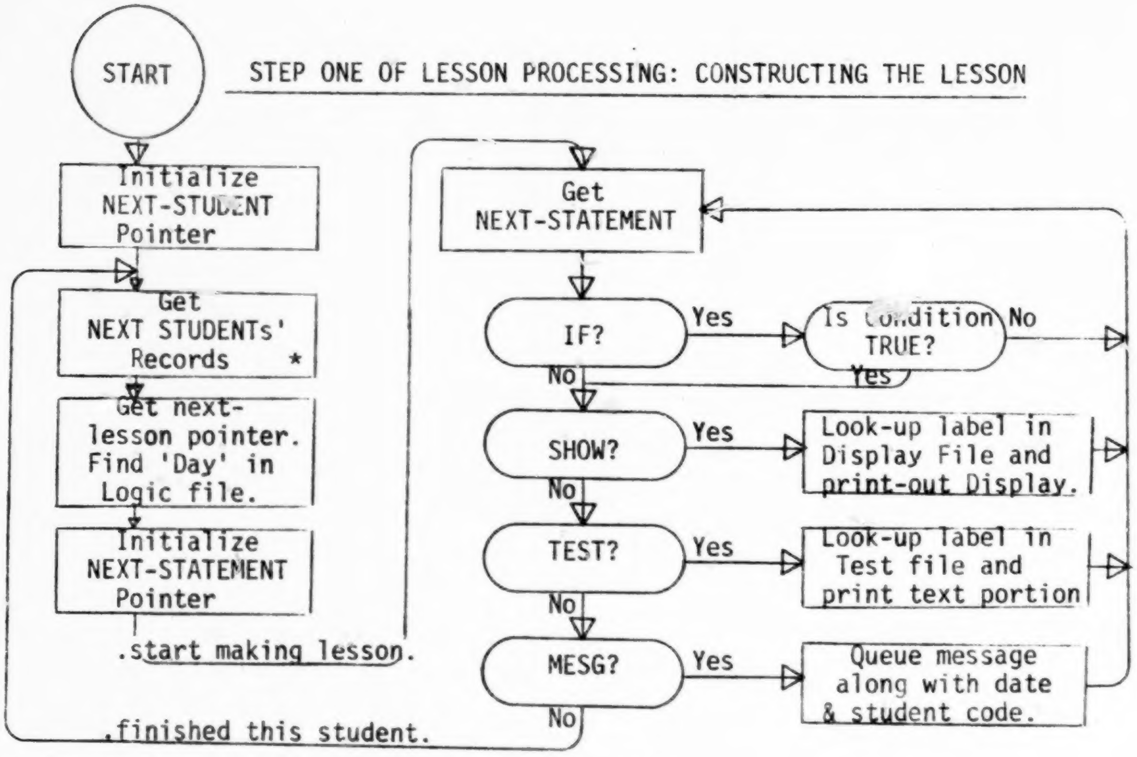
Online Capabilities

Under TAIM, there is one offline program (the overnight program) and a number of online programs. The term 'online' indicates that these programs run under the control of a user via a remote terminal. Where the offline program has a single duty (to determine and print a lesson for each student), the online programs are divided into the Mode 1 programs, the Mode 2 programs, and the Mode 3 programs.

Using Mode 1, the teacher may collect and print out data on the students and the four main files, and may alter the sequence of instruction for individuals or groups of students. The Mode 2 programs enable the teacher to modify the Logic, Display and Test files. The separation is a natural one; Mode 1 is used to run the TAIM system, and Mode 2 to alter its operation. The Mode 3 programs consist of caretaking and initialization routines. Their use requires an understanding of the architecture of TAIM; they will not normally be accessable to the teachers, and are therefore not discussed in this paper.

Mode 1 would be accessible to all teachers using TAIM. As soon as a user signs-on to Mode 1, he may issue any of thirteen possible commands. One of these, READ, runs a program to read-in student cards (discussed previously). The remainder can be divided into three sets: student monitoring, data gathering and miscellaneous programs.

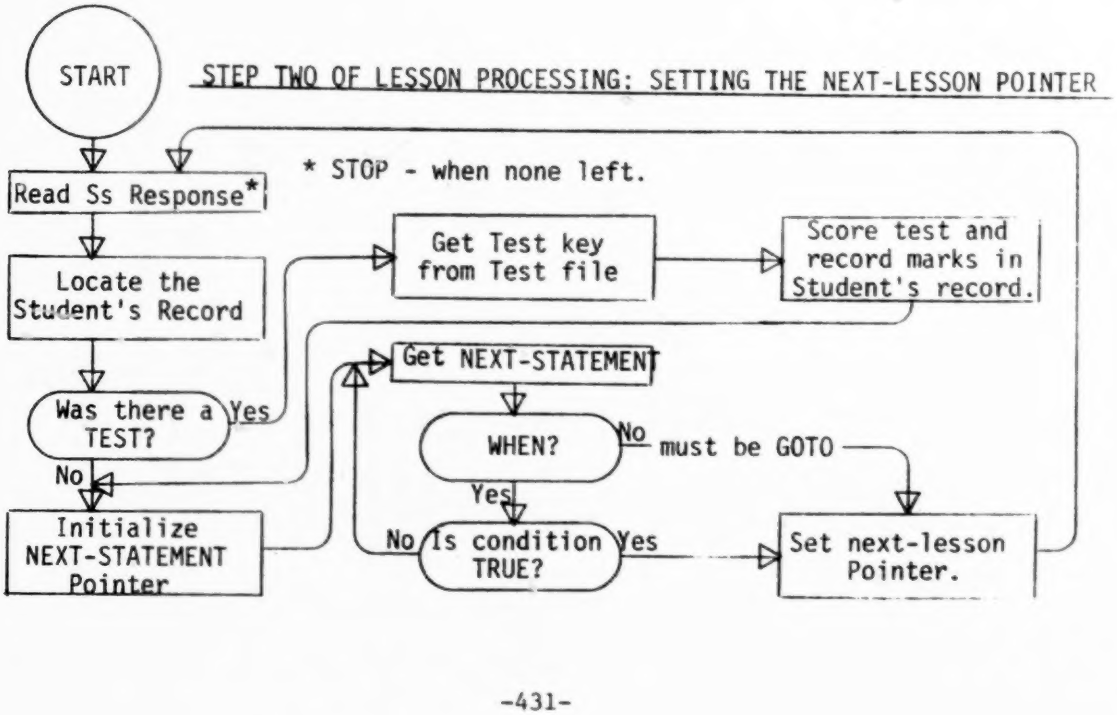
The student monitoring commands allow the teacher to alter the lessons assigned to the students by TAIM. The command NOTE allows



STEP ONE OF LESSON PROCESSING: CONSTRUCTING THE LESSON

START
Initialize NEXT-STUDENT Pointer
Get NEXT STUDENTs' Records *
Get next-lesson pointer. Find 'Day' in Logic file.
Initialize NEXT-STATEMENT Pointer
.start making lesson.
.finished this student.
Get NEXT-STATEMENT
IF?
Yes
Is condition TRUE?
No
Yes
No
SHOW?
Yes
Look-up label in Display File and print-out Display.
No
TEST?
Yes
Look-up label in Test file and print text portion
No
MESG?
Yes
Queue message along with date & student code.
No

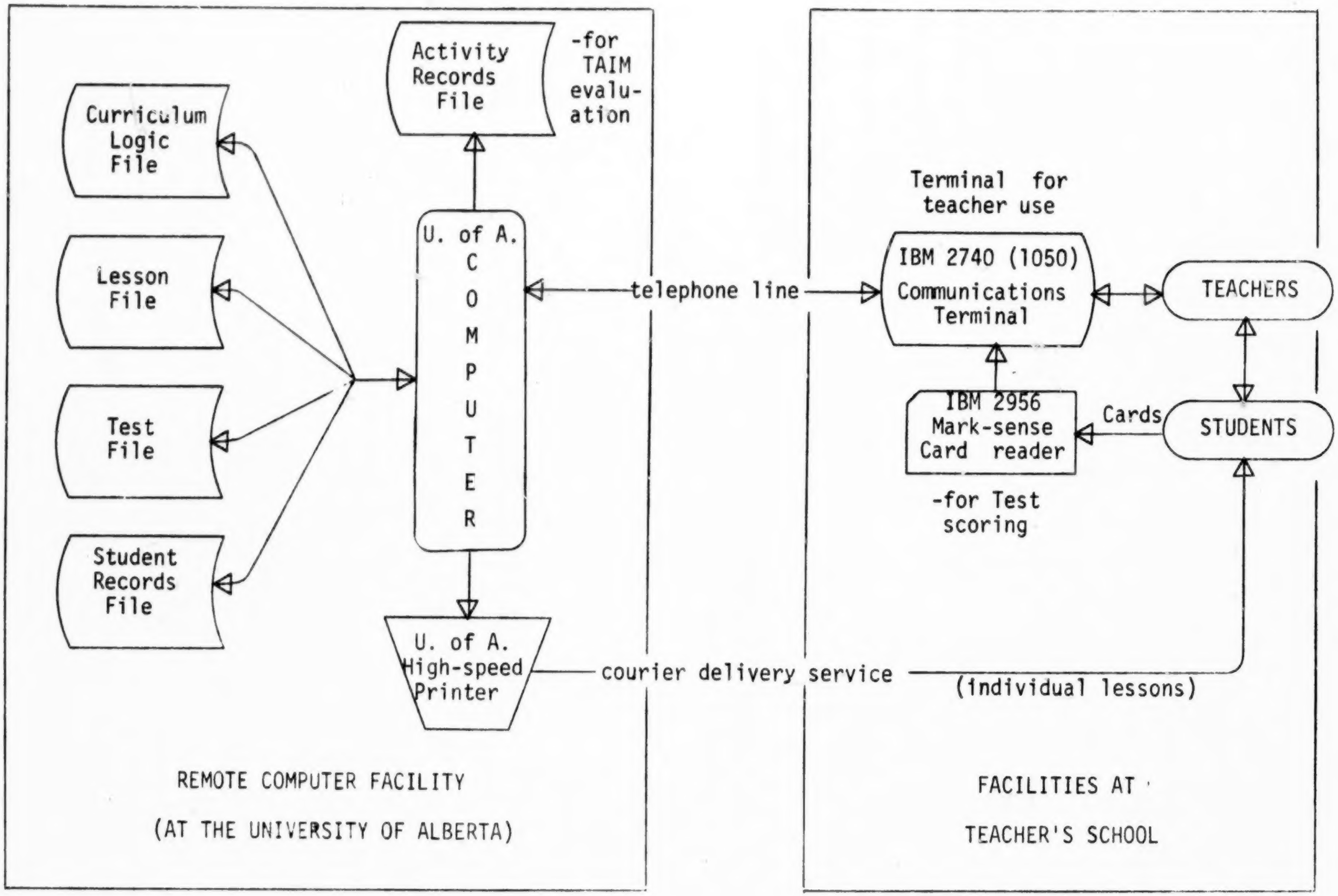
* STOP - when all students have been processed.



STEP TWO OF LESSON PROCESSING: SETTING THE NEXT-LESSON POINTER

START
Read Ss Response*
* STOP - when none left.
Locate the Student's Record
Was there a TEST?
Yes
No
Get Test key from Test file
Score test and record marks in Student's record.
Initialize NEXT-STATEMENT Pointer
Get NEXT-STATEMENT
WHEN?
No
must be GOTO
Yes
Is condition TRUE?
No
Yes
Set next-lesson Pointer.

GENERAL DESIGN CHARACTERISTICS OF THE TEACHER-AUTHORED INSTRUCTION MANAGEMENT SYSTEM


Curriculum Logic File
Lesson File
Test File
Student Records File
Activity Records File
-for TAIM evalu-ation
U. of A. COMPUTER
U. of A. High-speed Printer
telephone line
courier delivery service
(individual lessons)
REMOTE COMPUTER FACILITY
(AT THE UNIVERSITY OF ALBERTA)
Terminal for teacher use
IBM 2740 (1050) Communications Terminal
TEACHERS
STUDENTS
Cards
IBM 2956 Mark-sense Card reader
-for Test scoring
FACILITIES AT
TEACHER'S SCHOOL

the teacher to place a comment in any student's Student Record; this comment is then printed at the top of that student's next lesson. HOLD allows the teacher to suppress the lesson printouts for students. This might be used if a student was to be absent for several days, or if a group of students were to miss their regular classes for some reason. The POINT command allows the teacher to change any student's next-lesson pointer. This might become necessary if the decision algorithms have placed a student on a path which the teacher felt was not appropriate.

The data gathering commands allow the teacher to print out information from the files. There are four of these. PRINT is used for several purposes: to print the teacher messages generated by the MESG statements in the Logic file, to find out which students have been NOTEd, or are on HOLD, and to print selected portions of the Logic, Display, or Test files. The MARK command allows the teacher to retrieve the marks any student received on any past Test; the TRACE command permits the tracing of any student through the labels of any past Days; and the WHO command can be used to tell which students have taken a particular Day or Test.

Two other student monitoring commands are REGISTER, and UNREGISTER which simply add or delete Student Records from the Record file. The three miscellaneous commands are available in both modes 1 and 2; they are: MODE which allows the transfer between modes, STOP which simply stops the TAIM system, and WHY. Whenever a user issues a command, he gets a reply of either DONE or NOT DONE...#, where # is the number of an error message. The command WHY causes the message to be printed out.

Mode 2 will not normally be accessible to all teachers. This is because the modification of the Display, Test, and Logic files must be done very carefully - by someone who has a good grasp of the mechanics of the TAIM files.

When making up a Day for the Logic file, the teacher will typically think in terms of one day of instruction for a student. First, a presentation must be made; the student must be given some content. The teacher can examine the existing Displays to see if there is one or more available to suit his needs. If additional Displays, such as further explanation, directions for supplementary material, or a set of exercises are needed, these must be constructed, labelled, and inserted into the Display file. The teacher may decide that a quiz on the content is appropriate. If one is available, it can be used; otherwise a Test must be constructed, labelled, and inserted into the Test file.

With the necessary Displays and Test available, the teacher can begin construction of the Day with SHOW, TEST, and MESG statements.

The Day must have one or more exits to other Days which already exist. When these are chosen and the Conditions defined, the WHEN and GOTO statements can be placed at the end of the Day under construction. This Day must also have an entry; that is, one or more other Days must have possible exits to it. These must be set up by modifying one or more of the existing Days.

Care must be taken to ensure that all Days have both entries and exits, and that all units are properly labelled and contain meaningful statements. Failure in this regard would cause a number of students to not receive their lessons, or to receive improper lessons. However, mode 2 has a number of built-in checks to help eliminate these problems.

The mode 2 user is given five temporary workspaces where he can construct any units he wishes. He may place anything at all in these, but the only way to have them become part of the permanent files is through the SAVE command. This command runs a thorough check on all syntax.

In all there are sixteen mode 2 commands. These can be categorized as edit, data, evaluation, service, and miscellaneous commands. There are four edit commands: EDIT simply 'gets' one of the temporary workspaces, then INSERT, REPLACE, and DELETE can be used to add, modify or remove lines from it. There are two commands for the purpose of retrieving data from the files: PRINT can be used to print the contents of a workspace or a permanent unit, and FIND can be used to locate (in either temporary or permanent units) all references to a given label. For example; if REV3 was a Display label, then FIND REV3 would 'find' all Days which contained the statement SHOW REV3 and print out their labels.

There are four commands in the service category. PLACE simply places a copy of a permanent unit in one of the workspaces, and LABEL is used to assign a label to a workspace. The test key portion of each Test is in coded form; the command WEIGHTS causes TAIM to accept data from the teacher and code it properly for use with Tests. The command CHANGE is used to alter labels of permanent or temporary files. Note that if a label is changed on a permanent unit, then all references to that label must also be changed. CHANGE does this automatically. The three evaluation commands are: EXECUTE, EVALUATE, and SAVE. EXECUTE is used for semantical evaluation by the teacher. For example, a Test can be EXECUTEd; the teacher is asked for responses which are marked and printed out. The teacher can tell if the Test is being marked properly. EVALUATE does a syntactical evaluation, ensuring that all referenced labels actually exist, that each Day has entries and exits, etc. SAVE is the same as EVALUATE, except that if no errors are found during evaluation, the temporary workspace(s) are copied into the permanent files. The three miscellaneous commands are MODE, STOP, and WHY. These were described in the Mode 1 section.

Expectations

It should be clear that a teacher using TAIM will not have less work to do because of the system. For the first one or two years he would probably have more. One distinct advantage that TAIM can offer is to make the teacher's work cumulative both quantitatively and qualitatively. Consider the quantity of lessons a teacher prepares; there must be at least two hundred per class per year. Yet few if any of these are saved for use next year. Still fewer are transmitted to other teachers. Under TAIM, all would be saved and available.

However, quality is the more important factor (it is of little use to save poor lessons). Using current methods, a teacher about to present a lesson might attempt to recall how this or a similar lesson worked in the past. Identifying imperfections, he may try variations to improve the presentation. Clearly, the time to do such formative evaluation of a lesson is immediately after the presentation - not a year later. Under TAIM, this is not only possible, but should be defined as one of the teacher's main duties. Since the lesson is kept intact between evaluations, it is reasonable to expect that it will improve from year to year.

Another advantage of using TAIM is the system's ability to automatically store and retrieve information concerning individual students. A teacher constructing individual lessons may be able to say: "a student reaching this point should take either lesson 'A' or lesson 'B' depending on factor 'X'." Yet at some later time, the teacher will have difficulty both recalling the rule and locating information on the relevant factor. TAIM overcomes this problem by allowing the teacher to define the rule, saving it, and then automatically retrieving the required information and executing the rule whenever a student reaches the specified point. Further, the same sort of formative evaluation can be done on the rules as was done on the lessons.

The third distinctive advantage of the TAIM system is that it effectively eliminates a major portion of a teacher's tasks - those tasks which can be done by machine. This is expected to cause the teacher to focus on his unique talents: providing students with the human contact that cannot be simulated, and providing the supervision and continuous monitoring that the computer needs to remain adept at its job.

APPENDIX - ONLINE COMMANDS

MODE 1 USER COMMANDS

READ - reads in student cards

Student Monitoring

NOTE - place a comment in Student Record
HOLD - suppress offline printout of a student
POINT - alter student's next-lesson pointer
REGISTER and UNREGISTER - for student registration

Data Gathering

PRINT - print selected file contents, MESGs, NOTEs, HOLDs
MARK - print out student's Test marks
TRACE - print out student's past L-labels
WHO - print IDs of all students who have 'hit' a certain label

Miscellaneous

MODE - to change to mode 2
STOP - to get out of the TAIM system
WHY - to explain error messages

MODE 2 USER COMMANDS

Edit Commands

EDIT - to 'get' a temporary workspace
INSERT - to insert a line in a temporary workspace
REPLACE - to replace a line in a temporary workspace
DELETE - to delete a line from a temporary workspace

Data Commands

PRINT - to print out information from the files or workspaces
FIND - to locate all references to a given label

Evaluation Commands

EXECUTE - for semantical evaluation of units by the teacher
EVALUATE - for syntactical evaluation of units by TAIM
SAVE - to copy temporary workspaces to the permanent files

Service Commands

PLACE - to copy a permanent unit to a workspace
LABEL - to label a workspace
WEIGHTS - for automatic coding of Test keys
CHANGE - to change labels

Miscellaneous Commands

(see above under Mode 1)

A LABORATORY FOR CAI LESSON RESEARCH & DEVELOPMENT

Gary M. Boyd

A LABORATORY FOR CAI LESSON RESEARCH & DEVELOPMENT

SESSION 14B
ROOM/SALLE 145

By/Par: Gary M. Boyd,
Sir George Williams University.
Université Sir George Williams.

One promising approach for research and for the development of CAI lessons is to record, analyze and edit protocols obtained from actual tutorials.

A laboratory has been designed and constructed at SGWU with three student carrells and one tutor carrell. The tutor is linked to a student by closed circuit TV by a dual keyboard computer terminal (LEK III) and by an audio link.

Tutorials are recorded in computer files on the SGWU CDC CYBER70 (6400) system and on videotape.

The protocols are used to determine the number and variety of questions and answers required for efficient programmed instruction (curriculum and lesson design). It is hoped eventually to use the protocols as a basis for developing algorithms for generating problems and solutions.

The laboratory has been in operation since Dec. 1971 and tutorials are being conducted in aspects of English as a second language and on topics in the SGWU Humanities of Science programme. (Results of preliminary evaluation of programmes produced on the system will be given if available).

UN LABORATOIRE DE RECHERCHES ET DÉVELOPPEMENT POUR ENSEIGNER À L'AIDE D'ORDINATEURS

Une manière intéressante de se servir des leçons utilisant l'ordinateur, pour la recherche et le développement, consiste à enregistrer, à analyser et à éditer des "protocols" obtenus d'échanges réels.

A la SGWU un laboratoire a été construit comportant trois cellules pour les élèves et une pour le professeur. Ce professeur est relié à l'étudiant à l'aide de la télévision en circuit fermé par un double clavier à un terminal d'ordinateur (LEK-III) et par un ligne d'écoute. Les leçons sont mises en mémoire dans l'ordinateur CDC CYBER 70 (6400) de la SGWU et sur magnétoscope.

Les "protocols" servent à déterminer le nombre de questions et des réponses et leur catégorie pour avoir une instruction programmée efficace (la conception et les programmes des leçons). Nous espérons éventuellement nous servir des "protocols" comme base de développement d'algorithmes permettant de poser des problèmes et de les résoudre.

Le laboratoire fonctionne depuis décembre 1971 et les leçons se font sur les aspects de l'anglais comme deuxième langue et sur certains points du cours concernant l'aspect humain des sciences à la SGWU. (Les premiers résultats sur la valeur des programmes seront donnés si nous en disposons).

A LABORATORY FOR CYBERNETIC LESSON DEVELOPMENT

Gary M. Boyd, Sir George Williams University
April, 1972

One difficulty which arises when one attempts to introduce machine-based (CAI, teaching machine, PI) study materials into an existing educational institution is the difficulty of obtaining or producing instructional software which precisely suits the students' needs and the established curriculum of the institution. For the most part faculty are unable, or given the limited rewards available, undesirous, of producing their own programmes, and at the same time are very critical of materials produced elsewhere.

A potential solution to this problem is to produce study materials by transcribing and editing protocols obtained from person-to-person tutorials conducted by faculty with particularly needy, and with particularly deserving students.

Recording, editing and transcribing audiotapes or videotapes of tutorials is a difficult and tedious process. However, editing computer files using a contemporary large time-sharing computer system with a good edit system is very easy - easier in fact than using scissors and paste if any appreciable amount of material is involved.

One reason for setting up a computer-based tutorial recording

---

This project is funded by a grant from the Direction Générale d'Enseignement Supérieur Québec.

facility is to produce instructional programmes. A second reason for such a facility is to conduct research into the specific characteristics of programmes which enhance learning opportunities for particular types of students in particular subject areas.

Theories of instructional design (and teaching algorithms) tend either to be so general (.e.g. "provide immediate reinforcement") as to provide little effective guidance to the lesson designer, or on the other hand tend to be so specific that they have to be newly formulated for each lesson. In the case of the latter, a system which facilitates such formulation and the requisite testing is to be desired.

Given that recording and editing protocols may be a good way to produce validated lesson programmes and also may be a good way to develop teaching algorithms for specific classes of instructional problems, what sort of a facility is required?

The simplest facility would be simply a divided room to cut off direct verbal and non-verbal interactions of the sort which are too complex to analyze (or to be incorporated into machine programmes). A slot in the wall would enable tutor and student to exchange written and dtawn messages. Such a system is perfectly possible, but would require a lot of tedious clerical work and would limit the forms of stimuli more narrowly than available teaching machines or CAI systems.

Instead of a slot in the wall, if we give the student and the tutor each a computer terminal on a system wuch that both can communicate directly, then automatic transcription, timing and editing become possible.

In order to handle visual and auditory messages, it would be desirable to have computer terminals capable of scanning images and carry out analogue-to-digital conversions and store all messages in computer files. At present the cost of such sophisticated terminals and of the vast memory that would be required outweigh the benefits.

Audio-visual capability can be provided at reasonable cost by using closed circuit television as a link. A videotape recorder serves as an additional "memory", and by recording actual times from a digital clock on the videotape and in the computer files, it is possible to rapidly locate the segment of tape associated with any particular segment of computer-recorder dialogue.

Moreover, by using computer terminals which provide alphanumerice display in the form of a standard TV signal, it is possible to record the text along with the visuals on videotape and the whole facility can serve as an instructional TV production facility.

The general layout of the system is shown in Figure 1. Two-way communication between student and tutor is confined to the computer terminal keyboard channel, while one-way visual and auditory communication from the tutor to the student is provided by CCTV and an audio link. This constraint is necessary to enable initial editing and analysis to be carried out rapidly and easily using only the computer-stored data. It also corresponds to the limitations placed on student responses by most CAI systems and teaching machines which cannot accommodate freely structured verbal or graphic responses.

This facility differs from an ordinary CAI facility mainly in the provision for direct on-line interaction between tutor and student, and also in the provision of a closed circuit television-link.

The computer used to record and edit the protocols is a CDC 6400 computer with the KRONOS II time-sharing system.

The tutorials are recorded using a programme called SCRIBE which open files as needed, and records each expression typed by either the student or the tutor together with start and finish times and an indicator character (" for tutor; ' for student) to indicate who typed in the statement. A sample print-out of a SCRIBE file is given in Figure 2.

It is necessary for the users to type delineating characters at the beginning and end of their statements, which SCRIBE uses to call the time, and which serve rather like the expression "over-to-you" in a two-way radio conversation.

Once the tutorial is recorded, the experimenter goes through the record and deletes irrelevant remarks (such as "let's break for coffee now").

Then the tutor is asked to read through the file and identify errors and also to provide descriptive titles for each topic and sub-topic covered. Several tutorials covering similar topics may then be merged into one file.

The vetted file is then used as data for a programme called TRACON I, which converts the transcription into the format of a CAN CAI lesson. Each line is given appropriate statement numbers and blank labels and the "T" (text presentation) op code required by the CAN interpreter. A sample print-out of a TRACON output is shown in Figure 3.

Further editing is then carried out to insert tables of contents or "menus" of appropriate points, and the necessary CAN op codes to evoke proper branching.

Another conversion programme, TRAMITS, will be used to convert files into instructional programmes for the MITSI teaching machine. And the effectiveness of MITSI programmes produced in this manner will be compared with MITSI programmes written in the usual 'a-priori' fashion.

The videotapes materials can readily be converted into slides and tapes for use in MITSI or with a CAI system. It is also possible to incorporate slide or movie film sequences into the tutorial if the tutor desires to do so. A rear projection system feeds such materials up through the tutor's work surface to the TV camera so that he may point to details on the slides or films.

The one-to-one relationship provides for complex interaction but limits the number of students to be used as a test population. A response system and TV link to an adjacent classroom is planned to allow larger test populations, but it will greatly limit the complexity of responses and queries.

I call this a"cybernetic" lesson-development laboratory because it incorporates the three basic cybernetic principles:

1) use of feedback for goal attainment;
2) acquisition of requisite variety for control;
3) correlation of memories with respect to higher order invariants to achieve self-organization.

The provision of mutual feedback between student and tutor is obvious. The constraint imposed on the variety of message material to be introduced by the student on the one hand and the wide variety of materials at the tutor's disposal on the other is only one side of the requisite variety management problem. The other side of the problem relates to choice of a variety of students with abilities and debilities characteristic of the total population of learners. This latter problem has only been tentatively solved by choice of some of the best and some of the poorest students as participants for the tutorials, and more systematic sampling procedures will be developed in future.

The "self-organization" is organization of study materials (CAI or MITSI) programmes which is carried out by comparing the topics covered and the questions asked in a number of tutorials and consolidating similar texts. The retention of "peculiar" (i.e. unique) questions and answers will be determined by the frequency with which subsequent students refer to them.

The basic idea of the project is on the one hand to provide a simple and straightforward way for regular college lecturers to participate in the development of validated tutorial material at modest cost, and on the other hand to provide readily analysable tutorial protocols for research purposes.

References :

Boyd, Gary M. "Ueber die "notwendige Maechtigkeit des Repertoires" als Randbedingung in der Kybernetischen Padagogik" in
Rollett, B. / Weltner, K., Fortschritte und Ergebnisse der Unterrichtstechnologie, Ehrenwirth, Munich (1971)

Graf, Klaus-Dieter " Programmierter Dialog zur Erzeugung von Lehrprogrammen"
Z. Erziehungswissenschaftliche Forschung 4 (1970) 1, pp. 18-36

Shure, Gerald H. & Mecker, Robert J. "A computer-Based Experimental Laboratory" Admin. Sci. Quarterly (1962), 14, 286-293.

Smith, Roulette Wm. "Computer Simulation of Teaching processes" Doctoral dissertation Stanford University (1972) (This simulation is based on over 800 protocols obtained from human teacher-student tutorials.)

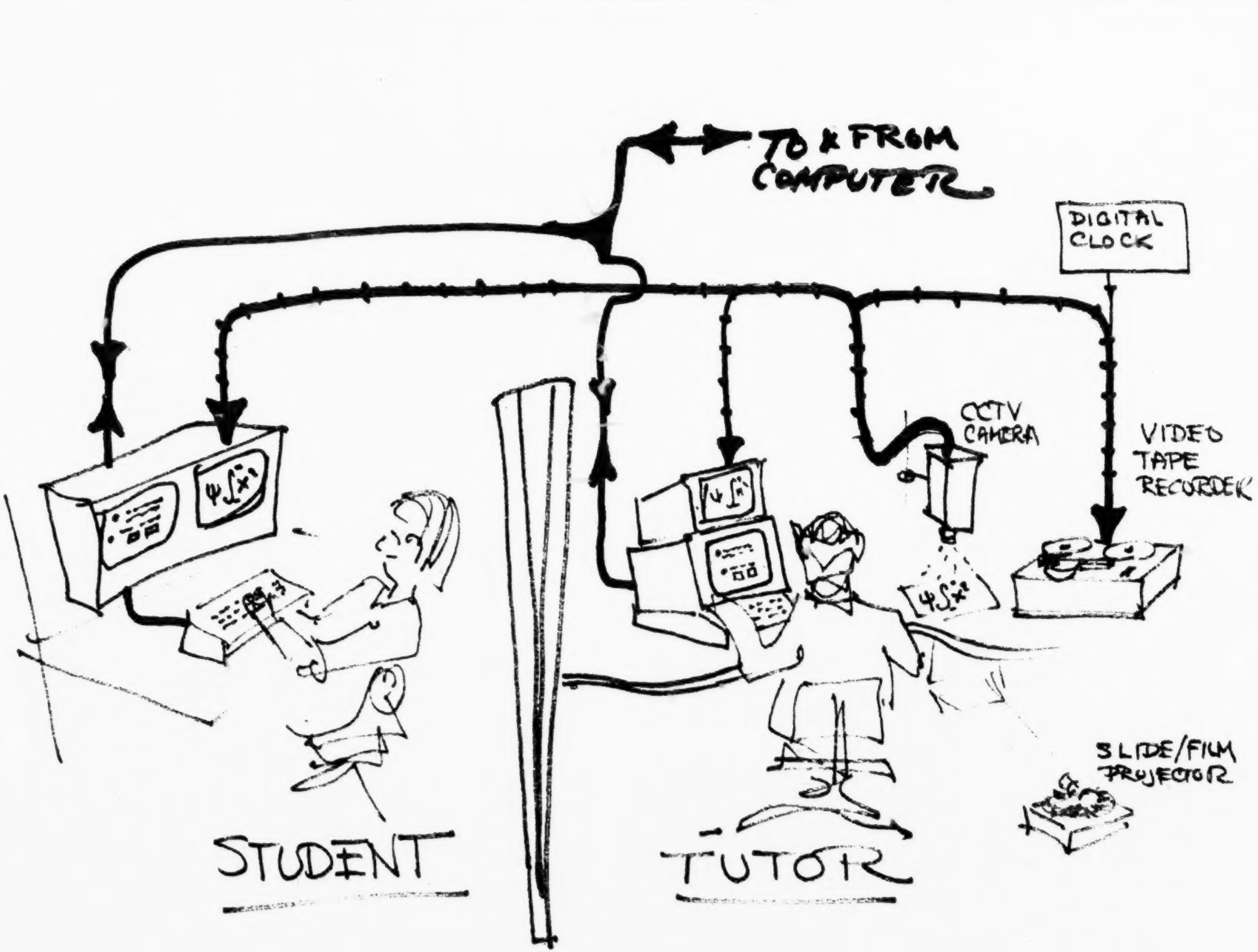

TO & FROM COMPUTER
DIGITAL CLOCK
CCTV CAMERA
VIDEO TAPE RECORDER
SLIDE/FILM PROJECTOR
STUDENT
TUTOR

FIG 1 LABORATORY ARRANGEMENT

G Boyd
March 72.

Fig 2.

Boyd
MAR 72

Beginning of a 'quasi' tutorial on 'Mediating' SCENE

```
**********
NEW RECORD       72/03/29. 10.58.45.
**********


TUTOR***** 10.59.13.

"VERY GOOD, THEN, LILY. WHAT
"SHALL WE DO THIS MORNING?

TUTOR***** 10.59.30.


STUDENT*** 11.00.07.

'WELL, COULD YOU PLEASE EXPLL
AIN TO ME ABOUT MEAZOLITHIC
PALEOLITHIC AND ALL THE OTHER
'

STUDENT*** 11.01.27.


TUTOR***** 11.01.47.

"FIRST OF ALL, WE MIST BE CERTAIN
WE KNOW THE DIFFERENCE BETWE-
EN THOSE WORDS ENDING IN
... 'ZOIC' AND 'ITHIC'. DO YOU
"KNOW THE DIFFERENCE?

TUTOR***** 11.02.43.


STUDENT*** 11.03.00.

'NO
'

STUDENT*** 11.03.05.


TUTOR***** 11.03.20.

"THOSE WORDS ENDING IN 'ZOIC'
(IE. MESOZOIC, CENOZOIC,ETC)
REFER TO GEO-BIOLOGICAL TIME
PERIODS. EITHER TIME PERIODS
WHEN WE ARE DEALING WITH THE
THEORY OF EVOLUTION 'BEFORE'
MAN, OR WITH GEOLOGICAL
EVOLUTION(AGE OF EARTH, ETC).
THOSE WORDS ENDING IN 'ITHIC'
REFER TO TIME PERIODS DEALING
WITH THE CULTURAL DEVELOPMENT
OF MAN. DOES THIS MAKE SENSE?
"

TUTOR***** 11.06.15.
```

FILMED FROM BEST AVAILABLE COPY

Figure 3. CAN SOURCE PROGRAM GENERATED BY TRACON.

```
OLD,STORE8B
READY.

LNH

0005 00000000 T
0010 00000000 T **********
0015 00000000 T NEW RECORD       72/03/29. 10.58.45.
0020 00000000 T **********
0025 00000000 T
0030 00000000 T
0035 00000000 T
0040 00000000 T TUTOR***** 10.59.13.
0045 00000000 T
0050 00000000 T "VERY GOOD, THEN, LILY. WHAT
0055 00000000 T "SHALL WE DO THIS MORNING?
0060 00000000 T
0065 00000000 T TUTOR***** 10.59.30.
0070 00000000 T
0075 00000000 T
0080 00000000 T STUDENT*** 11.00.07.
0085 00000000 T
0090 00000000 T 'WELL, COULD YOU PLEASE EXPLL
0095 00000000 T AIN TO ME ABOUT MEAZOLITHIC
0100 00000000 T PALEOLITHIC AND ALL THE OTHER
0105 00000000 T '
0110 00000000 T -
0115 00000000 T STUDENT*** 11.01.27.
0120 00000000 T
0125 00000000 T
0130 00000000 T TUTOR***** 11.01.47.
0135 00000000 T
0140 00000000 T "FIRST OF ALL, WE MIST BE CERTAIN
0145 00000000 T WE KNOW THE DIFFERENCE BETWE-
0150 00000000 T EN THOSE WORDS ENDING IN
0155 00000000 T ... 'ZOIC' AND 'ITHIC'. DO YOU
0160 00000000 T "KNOW THE DIFFERENCE?
0165 00000000 T
0170 00000000 T TUTOR***** 11.02.43.
0175 00000000 T
0180 00000000 T
0185 00000000 T STUDENT*** 11.03.00.
0190 00000000 T
0195 00000000 T 'NO
0200 00000000 T '
0205 00000000 T -
0210 00000000 T STUDANT

*TERMINAT
```

Boyd
March 72

THE CAI ACTIVITY OF THE DIVISION OF EDUCATIONAL RESEARCH SERVICES THE UNIVERSITY OF ALBERTA, EDMONTON

S. Hunka

SESSION 15
ROOM/SALLE 148

THE CAI ACTIVITY OF THE DIVISION OF EDUCATIONAL RESEARCH SERVICES THE UNIVERSITY OF ALBERTA, EDMONTON

By/Par: S. Hunka,
University of Alberta.
L'Université de l'Alberta.

The Division of Educational Research Services has operated an IBM 1500 CAI system for the past four years. The activities of the system can be categorized into three major areas: a) demonstration, b) research, and c) production.

The demonstration activities have involved a large number of lay and professional groups, teachers and students from local schools, and university staff and students. The research activity has been initiated from three sources: graduate students in various faculty departments, faculty staff, and the non-academic support staff for the system. Academic researchers have used the system more as a data collection device, while graduate students have used the system more in the study of computer assisted instruction.

During the past year the system has been used for the teaching of reading to beginning deaf children of age five to six years, APL statistical laboratories, Coursewriter programming to university students, and to provide an enrichment program to students in junior and senior high schools. It has been used to study problems in linguistics as well as in the measurement of intelligence through a simulation of the WISC.

The largest continuous production type of course which has been operated now for two years is the medical series in cardiology. This series has been the most successful in terms of student learning and attitudes. A second project has now been funded for the development of programs to simulate medical patients and some work has already been started in this area.

More recently a new research area has been opened by the funding through Canada Council of an oculometer project. This project is concerned with developing the hardware interface necessary to have a small mini-computer monitor the video output of a Honeywell video oculometer, and to calculate pupiliary dilation and the target being observed by the eye. The development is directed towards the study of eye movement and pupil dilation while a student interacts with the CAI computer.

TRAVAUX EAO DE LA DIVISION DES SERVICES DE RECHERCHES PÉDAGOGIQUES UNIVERSITÉ DE L'ALBERTA, EDMONTON

La Division des services de recherches pédagogiques a utilisé un système EAO IBM 1500 pendant les quatre dernières années. Ses travaux peuvent être répartis en trois domaines principaux: a) la démonstration, b) la recherche et c) la production.

De nombreux groupes de profanes et de professionnels, des professeurs et des étudiants des écoles locales et du personnel et des étudiants de l'université ont pris part aux travaux de démonstration. Les recherches ont été lancées par: les étudiants diplômés des divers départements de la faculté, le personnel de la faculté et le personnel de soutien "non-académique". Des chercheurs "académiques" ont utilisé le système plutôt comme dispositif pour rassembler des données tandis que les étudiants diplômés l'ont utilisé plutôt pour étudier l'enseignement à l'aide d'ordinateurs.

Depuis un an, le système est utilisé pour enseigner la lecture aux débutants sourds âgés de cinq à six ans, dans les laboratoires de statistiques APL, pour la programmation "Coursewriter" pour les étudiants universitaires et pour établir un programme d'enrichissement pour les étudiants des écoles secondaires. Il a servi à étudier les problèmes de linguistique et aussi à mesurer l'intelligence par une simulation "WISC".

Le cours qui s'est le mieux prêté à la reproduction en série pendant les deux dernières années est celui de cardiologie. C'est la série qui a eu le plus de succès si l'on considère l'attitude des étudiants et leurs possibilités d'apprendre. Nous avons des fonds pour établir une deuxième étude qui développera des programmes pour simuler des malades et on a commencé les travaux dans ce domaine.

Récemment, des fonds donnés par le Conseil du Canada ont permis d'entreprendre un autre travail, celui de l'étude de l'oculomètre. Cette étude a pour but de mettre au point la liaison entre un ordinateur miniaturisé et l'oculomètre Honeywell qu'il doit surveiller et aussi de calculer la dilatation des pupilles et la cible observée par l'oeil. Ce travail a pour but d'examiner quels sont les mouvements de l'oeil et la dilatation des pupilles lorsqu'un étudiant travaille avec l'ordinateur EAO.

THE CAI ACTIVITY OF THE DIVISION OF EDUCATIONAL RESEARCH SERVICES

S. Hunka
University of Alberta

The computer-assisted instruction facility at the University of Alberta has been in operation for about four years. During this time much has been learned about not only the technical aspects of CAI but also about the political and human factors which are involved.

The hardware configuration which we have is the IBM 1500 System. It was the first to be delivered outside of the United States, with the second one delivered to the Department of Education in Quebec. The system is basically driven with an IBM 1130 computer having 32K words of core, 1 system disk, and 4 additional disks for storage of course material. At the time of delivery only 8 terminals were available 2 of which were typewriters. A terminal is configured as a crt capable of holding 640 characters with half lines and column coordinates being addressable. The basic character is configured as a dot matrix of 8 dots wide and 12 dots high. Each terminal also contains a light pen, an image projector, and an audio play/record unit. About two years ago 8 additional terminals were delivered and a magnetic tape unit to permit transfer of material to the University's IBM 360/67 computer was obtained. For the past two years the amount of equipment has remained constant.

The software received with the delivery of the system consisted of Coursewriter II and MAT, the latter being a subset of APL. To the equipment and software we added three people: a manager, one programmer, and one operator, and an awful lot of faith. At that time the costs were $96,000 for rental of all the hardware and $16,000 for support personnel. The current costs are about $143,000 for rental of all hardware which includes 19 terminals, and about $55,000 for direct support of staff which include three operators, one system programmer, one Coursewriter programmer, and a manager. By commercial standards where personnel costs are taken to be roughly equal to hardware costs, we have much to be desired.

Delivered at the Canadian Symposium on Instructional Technology, University of Calgary, May 26, 1972

During the four years we have had the facility, no work to speak of has been done to actually improve the hardware capability of the system. Therefore, as far as this part of our work is concerned we have contributed nothing to the development of improved terminals or cpu hardware.

Contributions have been made to the improvement of software for the system. Perhaps most CAI authors are familiar with Coursewriter II, but only as it appears in regular IBM manuals. Although the basic operation codes are documented in such sources, the power of Coursewriter rests in its educational system support function and in assembler code functions which have been developed by ourselves and other 1500 installations. These functions were written because Coursewriter II did not provide the type of programming power required by programmers with even modest training. Many of the functions are directly related to problems involved in answer analysis and involve such operations as identification of keywords, keyletters, message synthesis, numerical limit functions, and so on.

More recently some important developments have been made in the construction and use of graphics or drawings which are displayed to the student on the crt. Initially, graphic units equal to the size of 3 characters high by 2 characters wide had to be constructed by punching each dot of light as a hole in a card. Recently, and within the context of Coursewriter II we have the capability of constructing graphics directly on line and visually observing what the graphic will look like when it is presented to the student. Authors may now use the light pen and construct drawings on their screens, concatenate drawings from one author's set to another's, rotate drawings around a horizontal or vertical axis, copy portions of one drawing to another, and re-position them on the screen. Near completion is a second facility which will allow the writing onto the crt screen of character strings magnified by a numerical factor specified as a parameter by an author. In the past we have had to spend inordinate amounts of time to have larger than normal sized characters available for young pre-school children.

Some changes have also been made to the operating facilities of the Coursewriter system. Different scheduling algorithms have been constructed to improve response time, and because of the inability to provide operators during the weekends and very late evenings, certain commands normally only available to operators have been made available to graduate student authors.

Although not directly connected with the 1500 system, we did develop a new authoring language which was a pre-compiler to Coursewriter II. This language attempted to separate instructional logics from subject matter in the same manner that in numerical programming one keeps the numerical logic such as in a Fortran source program, independent of the data. With the return of Dr. Romaniuk to our group, we hope that we will be reviewing this procedure. The availability of a large scientific computer, such as the 360/67 has been invaluable since the work of VAULT could not have been done on a small machine, nor could have a number of special programs for documenting Coursewriter II programs been written. All listings of code and documenting the technical aspects of programs is now accomplished through magnetic tape transfer of Coursewriter code to the larger university computer.

The operation of the 1500 system has been directed towards the attainment of three objectives: demonstration, research, and production. During the past four years there has been no problem in meeting the demonstration objectives since almost any CAI system with crts and some simple demonstration material can be made of interest to a wide range of people. Most frequent visitors to our installation have been teachers and their pupils. During the past four years there have been about 2000 public school pupils and their teachers visit the installation. Within the University itself we have had conducted demonstrations for about one quarter of the faculties, with the majority of demonstrations being conducted for students and their instructors in the Faculty of Education. Conducted demonstrations are also given to professional lay groups from the community and to instructors and students from other post secondary institutions. We have been very fortunate in that visitors have been very cooperative in the use of the system and no damage has resulted.

In the research area it is more difficult to describe in a meaningful manner what has been accomplished. In an academic institution one is tempted to cite the number of publications or theses which have been written. The Division has not been a prolific distributor of CAI publications because there have not been enough staff available to do this work, while other professors in the Faculty who have competence in CAI are sufficiently loaded not to have time to contribute to our work in this manner. A second factor is that the Division is not a teaching unit within the Faculty, and thus theses completed in the CAI area are the responsibility of the teaching departments. The departments most directly involved are Educational Psychology and Secondary education. In the former the system has been used primarily as a data collecting device for studies of learning, concept formation, visual search strategies, and more recently in simulations. In the other departments, concentration is directed more towards the study of instructional paradigms, with more attention being given recently to aptitude treatment interaction. In general, results have indicated that CAI is effective and a time saver. In most cases the programs constructed have not been optimized in the sense that they have been well tested and improved before an experiment has begun. This I suspect is too characteristic of CAI studies and is a factor which is generally not reported with the experimental results. In the most recent study completed by J. Isaacs, the CAI system programmed to teach logarithms using two instructional methods was compared with a controlled presentation within a classroom. Although CAI produced superior results relative to the classroom treatment, the program was one which was tested by getting other graduate students to 'try it out'.

There are major problems which have to be overcome before more research in the CAI area will become characteristic in universities. Academic staff members are generally sufficiently naive about computers that the CAI system is treated in the same manner as any other scientific number cruncher. There are few academics who are interested in trying to investigate the complex parameters involved in actual teaching. They appear more interested in more 'basic research' which from an instructional point of view, is frequently less relatable to CAI. In other words, we have few researchers who are willing and interested, and who have the resources to study instruction as an end in itself. Not an unimportant problem is that in our installation we have not been able to provide the kind of programming service which is required for a researcher not familiar with our authoring languages. The amount of time necessary to not only learn the authoring language, but then to program the material is sufficient to discourage most professors.

The basic back-up services required to operate a CAI facility which encourages academic staff members to carry out research is considerably more than what we anticipated in the beginning. Certainly the availability of grant funds are important, but probably not a sufficient condition. It is also necessary to have competent programming staff with members capable of communicating with academics. Taking teachers who have had one or two years of teaching experience and training them as specialized programmer for CAI seems to be an alternative if funds were available and particularly if there is a slight surplus of teachers on the current market. Further back-up resources are required for the preparation and integration of audio-visual, and special effects which are required by the authors. These services should not be scattered all over the campus requiring that the academic travel to several different technical service departments before he can get his work done. Further complications result when an author must do this laison work himself in that too frequently the technical producer does not fully understand the parameters which must be met in the production of the material. A second problem is that the producer of the audio-visual material does not see his product as it appears at the student's terminal.

We have had a number of projects which have been labelled by some as production and by others as research. The difference is not as clear as some would like, although I wonder sometimes why a distinction is required. During the last couple of years we have been fortunate to have a dynamic Faculty of Medicine where not only are new curricula being introduced but also newer educational delivery systems. Our largest and most continuous application of the CAI system has been in the instruction of second year medical students. This project has been a good application in the sense that it was a live experiment with 'real' students who took the course as a regular part of their medical curriculum. In addition, the subject matter author, Dr. R.E. Rossall, acted as a true author in the sense that he did no programming and in fact does not know Coursewriter, but had his instructional methods and subject matter handled by a non-academic programmer. This course has been extremely well received by over 200 students in undergraduate medicine, as well as interns and general practitioners, the latter using it as part of a continuing medical education series. In general the attitudes towards this type of instruction have been very positive and academic competence has been estimated to be better than in previous years when the lecture method was used. Some work in this area of cardiology has spread to pharmacology with a small amount of material available on the system during the latter half of this last academic term.

Two other projects involving real students have also been carried out during the last couple of years. One project involves a course entitled Introduction to CAI which does involve the teaching of CAI programming. Last year ninety percent of the programming was placed on the CAI system with the result that the students were able to produce as much as 10 times more coding for their individual projects than the previous year. A second project involves the use of APL for statistical laboratories. We have less evidence on the effectiveness of this work because the courses are not under the control of our own staff. In general instructors of statistics are not interested in integrating their instruction with the laboratories being completed on the computer. Frequently lab assignments which were originally designed for use with a desk calculator are given to the student to carry out using APL. No attempt has been made to use the file capabilities of the system for storage of common data or programs; no attempt has been made to use the system to simulate numerical properties of stochastic functions. I think little can be done to encourage 'real' use of the CAI system unless the academic sees some meaningful payoff to him. At the moment there still is little incentive for academics to increase their instructional efficiency.

During the last two years students who have worked closely with the Division have taken advantage of the Federal Opportunities for Youth summer employment program. Last year a student project was funded for the preparation of a program to teach the basic reading skills to deaf children who have just started grade one at the Alberta School for the Deaf. This project was funded for the sum of approximately $8000 and was used to formulate and complete the basic coding of the subject matter and instructional strategies. An additional $3000 was spent on the project in terms of indirect expenses incurred by the Division. The program systematically introduces the student to the use of nouns, verbs, objects, and adjectives with automatic review based upon a defined criterial level of performance. The program depends heavily upon the use of picture presentations. The most complex level required the student to construct small phrases of 4-5 words based upon his interpretation of a picture presented to him by the computer. Within the time period of about 8 weeks at roughly 1 hour per week, the children were able to discriminate over 100 words which they normally would learn during their full year in grade one. However, in checking their ability to match these words against the objects, there appeared to be little transfer. We have not investigated the monitoring begins by the computer through the use of Schmidt triggers which are adjusted to identify slight voltage changes which occur when the retina and the corneal reflect the infrared light. It is expected that the hardware interface will be completed by late this summer. One expected spin-off of the oculometer should be that a paraplegic could control our CAI system.

What does the future hold for us? Much of our future rests with the future of universities in general. Because of the stringent budget problems which have been encountered for this year, the University is planning to purchase the IBM 1500 system in order to reduce its operating budget. In this plan we hope to change our 1130 cpu for the fastest model which operates at 2.2 microseconds and to purchase disks which are 10 times faster than the current set of disks. Therefore, we shall remain in the CAI area for at least the next five years. We are not sure what the future holds for us in terms of support personnel which have always been required. Over the last four years we have had only one academic within the Division who was interested in CAI although there have been a large number of graduate students wishing guidance in this area. If further budget reductions are required in the year 1972-73, and with the purchase of the system about nintey percent of our funds will be for staff, further budget reductions would mean staff reductions. The current shortage of funds, coupled with the withdrawl of NRC block grants to computing centers has encouraged our computing center to institute a charging system. The charge of $1.50 per connect hour plus cpu time at $300 per hour is not going to encourage anyone to use the 360/67 as a CAI system.

In terms of the growth of CAI in the province, much is resting on the report being prepared on the future of education by Dr. W. Worth. Government officials and ministers are holding back any major decisions until it is released some time next month. This report will require careful study since some of the most important decisions regarding the use of CAI will be made at a political level not at the academic level.

the psychological differences between the testing methodology employed by the school and that employed with the CAI system. A similarly sponsored program has been funded for this summer, but the subject matter is French. In this project an attempt will be made to develop a program for instruction in French for elementary and junior high school children. Further information may be obtained by writing Mr. Victor Muller, c/o of the Division of Educational Research Services.

Two other projects have been in progress which involve students from local schools. One group of students from junior high schools in Edmonton have learned to program in Coursewriter and developing small programs to be used in their own classrooms. Another group of high school students are working entirely on their own using the system for APL work. Unfortunately, we were unable to find an academic to help them with their work. One thing we learned early in these two projects is that if one is not careful the students will take your system over.

Our most recent and only venture into the area of hardware development is the oculometer project. This project was funded by Canada Council. Essentially we are attempting to develop an infrared oculometer which can be used for tracking eyemovements and to calculate pupil diameter. When operational this system should allow us to rapidly and accurately follow the eyemovements of children watching the crt screens of our CAI system. In this way we think we will be able to obtain a better method of evaluating screen displays and material being presented.

The equipment consists of a standard PDP-8 DEC computer and a Honeywell oculometer, analogue to digital and digital to analogue converters, as well as magnetic tapes to store the volume of data which is expected. The video output of the television monitor is fed to a special hardware interface constructed by Professor Milton Petruk of our Division. This interface uses the horizontal pulse generated within the monitoring video camera as a triggering pulse to reset crystal clock counters used to determine the time necessary for the scanning beam to encounter the pupil and the corneal reflection, as well as to reset the line scan number which is generated within the camera. Software within the control computer is used to determine whether an odd or even line is being scanned since the video camera scans on an interlace pattern. Once all initialization has been completed

ASKING THE RIGHT QUESTIONS ABOUT INSTRUCTIONAL APPLICATIONS OF THE COMPUTER

Edwin F. Taylor
Senior Research Scientist, Department of Physics
and Education Research Center
Massachusetts Institute of Technology

ASKING THE RIGHT QUESTIONS ABOUT INSTRUCTIONAL APPLICATIONS OF THE COMPUTER

SESSION 16
ROOM/SALLE 148

By/Par: Edwin F. Taylor,
Massachusetts Institute of Technology.

Everyone interested in education who has access to a computer is tempted to ask the wrong question, namely, "Given this particular computer facility, what can I do with it in education?" The answer is a trap: "Almost anything, but...". The trap is that most educational uses springing from this naive interchange quickly exhaust both time and money, and typically lead to rigid systems of instruction that treat the student as a slave "for his own good". The original question was both too limited and too comprehensive; it now seems more productive to split this question into two parts. The heroic part starts at the computer end: "What is the optimum design for a comprehensive educational computer system?" In effect, all questions of design, cost, educational practice, and utilization are asked simultaneously. An example of the heroic effort is Project PLATO at the University of Illinois. The probability of success is not proportional to the size of the effort, and many such projects should be funded. The second part of the original question starts at the educational end and asks only the next question in sequence: "What technical or other aid is best for the particular learning problem (of either style or content) faced by my students at this moment?" Usually the answer will not involve computers, but occasionally they will offer a brilliant solution, such as the wide use of an easily-programmed computer as calculator (Dartmouth and elsewhere), and the computer as animator of film loops that model physical systems (MIT and elsewhere). Some ways of organizing courses allow piecemeal testing of such technical aids. The combination of chipping away at the use of computers from the educational end and blasting from the computer systems end promises a good mix of short-term and long-term payoffs.

LES BONNES QUESTIONS À POSER SUR LES APPLICATIONS DES ORDINATEURS DANS L'ENSEIGNEMENT

Tous ceux qui s'intéressent à l'éducation et qui ont la possibilité d'utiliser les ordinateurs sont tentés de poser une fausse question: "qu'est-ce que je peux faire avec cet ordinateur en pédagogie?". La réponse est un piège: "presque tout, mais...". Le piège réside dans le fait que la plupart des applications pédagogiques résultant de cet échange primitif épuisent très vite le temps et l'argent; généralement il mène à des systèmes rigides d'instruction dont l'étudiant devient l'esclave "pour son propre bien". La question posée à l'origine était à la fois trop restreinte et trop vaste; il nous parait maintenant que, si on divise la question en deux, le rendement sera meilleur. La partie "héroïque" commence à l'ordinateur: "quelle est la meilleure conception pour avoir un système global pédagogique à l'aide d'ordinateurs?". En effet, toutes les questions de conception, de prix de revient, d'habitudes pédagogiques et d'utilisation sont posées en même temps. Un des exemples de l'effort "héroïque" est le projet "PLATO", à l'Université de l'Illinois. La probabilité de succès n'est pas proportionnelle à l'effort et beaucoup de ces programmes devraient être subventionnés. La deuxième partie de la question d'origine se situe dans la partie éducative et ne pose que la question suivante dans l'ordre: "quel est le meilleur moyen, technique ou autre, pour résoudre le problème particulier (problème de style ou de contenu) auquel mes élèves font face en ce moment?". Généralement, la réponse n'implique pas l'utilisation des ordinateurs mais, de temps en temps, ceux-ci peuvent offrir une solution brillante, par exemple, l'utilisation étendue d'un ordinateur, facilement programmé, comme calculateur (à Dartmouth et ailleurs) et comme animateur de films en boucles représentant des systèmes sous forme de modèles (à M.I.T. et ailleurs). Certaines méthodes d'organisation des cours permettent d'essayer, partie par partie, ces moyens techniques. Si les pédagogues se servant d'ordinateurs et acceptant de faire des observations constructives, travaillent en collaboration étroite avec des ingénieurs soucieux de perfectionner les ordinateurs, on devrait arriver à des solutions avantageuses, à court terme et à long terme.

TWO WRONG QUESTIONS

Two great traps in the instructional applications of the computer are localism and empire building. Localism seeks an apparent economy: to use whatever computer facility is locally available to do something useful in education. It fails to recognize that the rigidities of all but a few of the most expensive standard computers render them wholly inadequate to the subtleties of instruction. Empire building looks to organization to bring the benefits of computer assisted instruction to large numbers of students. It fails to recognize that there is as yet no agreed-upon benefit of CAI that can be conveyed to large numbers at reasonable cost. Ours is the age of the pioneer; the age of the entrepreneur is just dawning. We look more closely at these two traps.

Localism asks the question: Given this particular computer facility, what can I do with it in education? (the first wrong question). The answer is a trap: "Almost anything, but ...." When powerful facilities are available, the "almost anything" part of this answer is impressive. At MIT we once set up a "conversational" system that carried on (via typewriter) a coherent dialogue on several subjects from physics to poetry with never an embarrassment or loss for words.[1] I have heard proposed a system in which an extensive structural model of "reality" in some field is compared with a continuously updated model of a student's understanding of this field, the computer ranging over both structures probing and improving the student's understanding with questions and instructions assembled from the data bases. Interactions with graphic displays can be delightful and powerful guides to intuition, particularly displays developed at technically sophisticated installations such as the Livermore Laboratory in California.

The "but ..." part of the answer ("Almost anything, but ....") is equally impressive. Our conversational dialogue system cost fourteen dollars per hour terminal time plus processor and file space, took one man-year to systems-program and one man-year to apply to a dialogue that occupied less than one week of a student's time. The "reality model" tutorial would require much more programming and a half-million words of dedicated memory. The Livermore interactive display projects exist as sideshows in an immense military development center.

Facilities available to most of us (particularly, in the U.S.A., since funds dried up) constitute a restricted temptation to localism. One classic quandry concerning limited facilities is whether to spend small resources developing a new and better computer language or to use whatever language is available in order to get on with the job of education. Alfred Bork and Richard Ballard describe the usual results:[2]

> These developments have tended toward one of two fates. Firstly, those producing new languages have often exhausted their resources and energies short of producing sufficient teaching materials to establish convincingly the merit of their approach. Those seeking to produce complete courses of computer-assisted material have tended to accept the language imposed limitations, opting for simple multiple-choice, limited response, and numerical branching structures, in order to achieve the necessary volume. The resulting programs only occasionally exhibit educational strategies or facilities distinct from those found in conventional or programmed instruction.

Similar fates await most of those who design terminals or devise communications systems for education. It is usually unproductive to nibble away piecemeal at the separate technical innovations that must be orchestrated in a successful educational application of computers.

Empire building asks the question: How can we organize to apply existing techniques on a large scale? (the second wrong question). A look around quickly exposes "existing techniques" to be perplexingly many and varied. Moreover, we wish to escape localism and are cautioned by those who administer and fund our work to make use of the most effective system available. So we fall into a further trap: we take a survey. MIT receives numberless requests to fill out forms describing our educational computer work. These forms end up in the wastebasket because (1) they typically force all descriptions into a single mold to allow classification, thus suppressing uniqueness, (2) developments are so rapid that most projects are transformed or discontinued before survey results are published, (3) in a published document one cannot be candid about failures, and (4) almost (but not quite) all education experiments that do not fail outright yield no significant difference when compared with conventional instruction. ("In the dark all innovations are beautiful"). Leave surveys to the professionals; Karl Zinn is at this conference.

TWO RIGHT QUESTIONS

We need to carry out investigations in instructional uses of the computer that have some chance of proving useful in the long run while avoiding localism and empire building. I believe we have seen enough failures and potential successes to recognize at least two fruitful styles for such investigations. The description of these styles will sound strangely like localism and empire building and indeed are the "good sides" of these two perversions.

I believe that at the present epoch the problems posed by instructional uses of computers must be chopped firmly into two parts. The heroic part starts at the computer end: What is the optimum design for a comprehensive educational computer system? In effect, all questions of design, cost, utilization, and educational practice are asked simultaneously and from the beginning. We need the Henry Fords of automated instruction. Henry Ford is credited with applying the assembly line to automobile manufacture, but his accomplishments also encompassed invention, development, organization, finances, merchandising, labor relations, and (abortively) international politics. He did not, of course, take any responsibility for the ultimate problems of public policy posed in our day by wide use of the automobile; he could no more predict these problems than we are able to envision the ultimate consequences of computer use in education. But he did become entrepreneur and, yes, empire builder by virtue of his willingness to engage simultaneously all questions about establishing the automobile on the roadways of the world.

Please understand: answering the heroic question about an educational computer system does not require spending a lot of money, at least initially. Project PLATO at the University of Illinois expects someday soon to install 4000 terminals around the state and into Canada. This is a heroic effort with, I believe, a strong promise of success. Yet for most of the years of its existence, until 1967, the project had no more than four full time equivalent employees and an annual budget of $150,000 or less (plus donated computer hardware).

A second fruitful approach to the instructional use of computers starts at the education end and asks only the next question in sequence: What technical aid or way of organizing learning is most effective for my (small scale) educational problem? The focus of attention is the educational need and not a pre-existing computer facility. Indeed, most often the answer to the educational question will not involve computers, but sometimes they will offer a brilliant solution. The question implies a running knowledge of educational innovations (computer-based and other)--but not from indiscriminate surveys! One needs to monitor the literature and to obtain good advice on a personal basis from those with taste, style, and discrimination in educational change.

Whereas the heroic effort to design an entire system can be likened to a full scale invasion into enemy territory, the educational question is a guerilla attack or commando raid. Two such commando raids that have been very successful for us are the generation of film loops and the use of the computer as computer. We have used computers to animate films that model physical results in realms that are far from everyday experience.[3] For example, in physics the subjects of relativity and quantum physics deal respectively with the very fast and the very small, both of which regions are described well by mathematical laws that are very hard to understand intuitively. A display that models the results of these laws with appropriate scaling constitutes a kind of demonstration that can focus and summarize understanding while eliciting and directing questions. Notice several features of producing film loops that make such projects fruitful: (1) One does not mind spending money once in order to produce the master print that can then be reproduced inexpensively. Various commercial and academic centers offer technical facilities and staff to program, display, and film these sequences. (2) Dissemination is simple and relatively inexpensive by means of film loops or regular 16 mm film. (3) We have found that the loops are so rich in physical insight that film titles cannot possibly explain all features of the display. In consequence we have begun to produce study booklets and other aids that can accompany and interpret these loops. (4) Planning, programming, and producing a film loop makes an excellent student project. The student undertaking such a project must master the professional content of the film as well as the technical aspects of execution and the educational uses of the result--all within the context of a limited and specific goal.

Another successful small scale educational use we have made of computers is as computing slaves for students. Student exercises in science and engineering have too often provided a limited experience with over-simplified special cases in order to make it possible for the student to carry out calculations himself. With access to a computer he can undertake very much more difficult problems and explore them much more widely by varying parameters and his method of attack. Along the way the student encounters this delightfully ironic twist: he must formulate with precision what he wants the computer to do and the computer executes his instructions literally--what better way is there to learn than to become the teacher of a facile-but-stupid machine? In using the computer this way on an occasional basis, we have sometimes found it economical to hire commercial time-shared BASIC systems rather than using the more expensive MIT computing facilities that are unnecessarily powerful for our purpose. Dartmouth College has been in the forefront of this effort, solving simultaneously the problems of computer language, hardware and software, integration into educational practice, communication, funding, and dissemination on its own campus and to nearby campuses and high schools, thus elevating the enterprise to the level of a heroic effort.

Some innovative methods of organizing a course can provide a setting for the piecemeal application of small educational experiments. For example, we have made considerable use of the so-called Keller Plan (sometimes called self-paced study or the personalized system of instruction) according to which each student can progress at his own rate through self-study materials with a lot of personal help from student tutors and tests of mastery at each step in his progress.[4,5] The Keller Plan is not primarily computer-based, but we have taken one or more units of study in such a course to try out film loops and the use of computer as computer (as well as such non-computer aids as take-home experiment kits, demonstration experiments, and programmed supplements). Since students pass through a particular study unit at different times, it is possible to revise the material continuously in the light of experience or to replace it entirely if it fails with the first students who reach it. Thus self-paced study provides a fertile test bed for many small educational innovations including computer aids.

Chopping the computer-education business apart into the heroic enterprise and the careful appropriate use of partly-proven techniques helps us to see where our individual efforts can be of the most use. It leads to the really deep question, "Who are you?" If the answer is, for example, "educator and physicist" then the oracle will have no difficulty in advising us: "Then educate in physics by whatever means are at hand ... leave the Henry Ford projects to others."

A SMATTERING OF OTHER QUESTIONS

Here is a random list of additional questions assembled in the process of composing this paper. These questions were all asked and answered, implicitly or explicitly, for every one of our successful uses of computers, and should have been asked but were not in all of our failures. In the Appendix is a further list of questions published previously to accompany an analysis of the failure of a computer interactive display.[6]

WHO will tell you that you are crazy? Someone will always volunteer this opinion, but it is important to solicit advice from the right hard-nosed commentator. Professor Victor Weisskopf, a world figure in physics, was shown the conversational tutorial system described above when it was still being developed. In less than two minutes he recognized that the analysis of student conceptual errors was better handled by other humans than by machines, and said so clearly and forcefully. We finally came around to his point of view only after expending many further man-hours and foundation dollars on the project.

Can students vote with their feet? A very effective evaluation method is to prescribe as concretely as possible what you expect a student to be able to do and then to give him a computer aid as one of several alternative tools for developing the required skill. Our educational system carefully trains students to find the most efficient path to a prescribed goal, so their choice of a computer program will be an honest vote in its favor.

Does the program ask computers to be people? Although I know of no theoretical limit to the computer's ability to mimic human intellectual processes, there reamins a world of difference between a CDC 6600 at $500 per hour and an undergraduate tutor at $2 per hour. In our experience the undergraduate is far and away more capable of identifying difficulties a student is having than the presently most sophisticated computer program. Taste, judgment and analyzing errors are, for the present, better left to people.

Does the program ask people to be computers? Here one all too often recognizes the 3M Distemper: memorization, manipulation, and multiple choice. Each of these may have their place at some stage of education, but almost without exception they are more efficiently carried out through more established technologies.

Who is the master? The crucial question toward which our line of thought has been leading. We abandoned our conversational computer tutorial program, in part, because at any stage of a dialogue the computer had to be able to identify the context. The best way to do this was to have the computer ask questions; and it was our final judgment that a computer has no business bossing students around. The Dartmouth BASIC system, available to all students, is educationally productive in direct proportion to its powerful and facile slavishness.

How will you recognize a failure? Education experiments have a way of succeeding by definition, whereas in fact one success in five experiments is as good an average in education as it is in the laboratory. John Archibald Wheeler defines genius as the ability to make all possible mistakes in the shortest possible time. With excellent advice and continuous thought, you may be able to try one promising idea after another so rapidly that successes come more often than your colleagues can recognize the failures. That is called research. But should you get locked into an expensive and unproductive enterprise, bury it as quickly as possible. Then, after the shock has worn off, let the learning juices flow around the experience. Together with what you learn from success, it may qualify you to give invited papers at conferences in beautiful surroundings far from home.

APPENDIX

QUESTIONS TO ASK ABOUT A PROPOSED TECHNICAL LEARNING AID[6]

We list some questions one might ask about a proposed technical system that is designed to fulfill or assist in fulfilling some educational purpose. The list does not include (and should be added to) questions about the professional correctness of the materials, the strategy of their presentation, the resources necessary to develop the tool, the competence of those making the proposal, long-term effects on the curriculum, and the human effects of the resulting innovations.

1. Does this device truly teach anything?

2. What does it teach?

3. How well does it teach compared with already-existing methods or simpler alternative methods that might be developed?

4. Have students been involved in the planning for this device?

5. Have preliminary or mock-up versions been tried with students?

6. How do students respond?

7. How much of his own money would a student be willing to spend to use the device for one hour? (Ask him!)

8. Are local faculty who might use the device participating in its design and execution?

9. Is it based on an established technology, defined as one for which professional commercial trouble-shooting and repair is quickly available locally?

10. If maintenance and repair are to be provided by the school, what is the average annual salary of the staff person needed for this purpose, including fringe benefits?

11. Can an adaptation of the tool to a more established technology allow much wider dissemination with only slight sacrifice of its central education features?

12. Can the new tool be used in courses as presently run? If it is successful, in what direction will it influence the development of these courses?

13. What publicity is required to encourage full use of the device?

14. How many students can use the services of this device simultaneously?

15. How many hours will each student use the device in the course of learning what it has to teach?

16. What is the cost per student hour, including overhead?

17. What is the marginal cost if one student uses the facility for one additional hour?

18. What is the additional capital cost, above present investment, to provide the service to a class of 10 students? 100 students? 1000 students?

19. If a school 100 miles away wants to install this system, what will be required in professional advice, in staff at the new location, in capital investment, in running costs per student hour, and in maintenance and repair?

REFERENCES

1. Edwin F. Taylor, ed., ELIZA: A Skimmable Report on the ELIZA Conversational Tutoring System (Cambridge, Mass.: Education Research Center, Massachusetts Institute of Technology, 1968).

2. Alfred Bork and Richard Ballard, The Physics Computer Development Project (Irvine, Calif.: University of California, Irvine, 1972).

3. Judah L. Schwartz and Edwin F. Taylor, "Computer Displays in the Teaching of Physics," *AFIPS--Conference Proceedings, Vol. 33* (Washington, D. C.: Thompson Book Co., 1968).

4. Ben A. Green, Jr., "Physics Teaching by the Keller Plan at MIT," *American Journal of Physics*, vol. 39, no. 7 (1971), p. 764.

5. Fred S. Keller, "Good-bye, Teacher ...," *Journal of Applied Behavior Analysis*, vol. 1, no. 1 (1968), p. 79.

6. Edwin F. Taylor, excerpted from "History of a Failure in Computer Interactive Instruction," *Conference Proceedings--Computers in Undergraduate Science Education* (College Park, Maryland: Commission on College Physics, 1971).

Closing Summary

A.D. Booth
University of Saskatchewan

It is an almost impossible job to sum up this conference, I thought so before I came to this last session but, after listening to Dr. Taylor, I am quite sure. I am going to offer you some comments in the hope that you can use them as a stimulus. Before I start, however, I want to mention that the Associate Committee which organized this conference with the assistance of the University of Calgary is doing two things about publication: one is to publish the "Proceedings". The other is that the Associate Committee approved the idea of publishing selected items from the Proceedings in the International Journal of Man Machine Studies. I shall be editing this and I request either that I write to some of you to invite submission of lengthened versions of your papers, or that you send in papers to me as quickly as possible. For Dr. Hunka I might remark that this is a good way of accumulating a few "Brownie points", I was looking hastily at the notes for authors in the back of the J.M.M.S. and noticed that there is no page charge and that you get fifty free reprints. So this is an even better investment for the future. A further comment on Hunka's remark is that the notion of making page charges seems to be an entirely American idea. I never publish in journals that make a charge, there are plenty of journals that will pay you, and in this way you can get two lots of Brownie points, a little investigation will show authors which Journals are involved. I start by making two remarks on Dr. Taylor's powerful address, they are in a sense anecdotal. The first is in connection with his observation that the students use of the computer gives an honest assessment of its worth. I would like to challenge this. I give, a fourth year class on Mathematics for Engineers. One of the things that the students imagine from my background is that this class is one which involves the use of the computer. I point out to them that any example set may or may not involve the use of a computer and that if they use a computer on an example which does not involve computer use they will be penalized, not only for that example but for the whole course. I regret to say that, in the last batch, which is fairly typical, the whole group of 43 students all used computers on all of the problems. None of the problems were supposed to be done this way at all! So I am not in the least bit sure that use of the computer indicates anything at all except, perhaps, student stupidity. I have got another suggestion for Dr. Taylor, as he invited instant comment. This concerns what he does with his children on a Saturday morning to stop them looking at the T.V. cartoons. I have a very easy solution to that which I in fact use: don't have a T.V. set.

Now let us get to more serious matters. One comment I heard at the beginning of this set of meetings, was to the effect that some people were disappointed that there was no extensive presentation of the latest goodies in AV-T.V. I disagree, these are state of the art items which you can see all about you, particularly in this well endowed university which has everything from closed to projection T.V., things which are not often found in schools. It is particularly nice to be able to complement the University of Calgary on this, in view of Dr. Hunka's remarks on the poor financial state of Alberta. This leads rather naturally to the comment that Dr. Taylor made on the piecemeal approach to the solution of problems versus the apparently well thought of heroic approach of looking at the problem as a whole. I think this is worth singling out for special comment and I would like to commend to you the following. In Canada most of us do not have the funds that MIT have, even in their present state of "financial distress". Thus we cannot engage in heroic experiments. I think that much can be done by devoting some time to thought before starting on experiment. Such thought may define the system so that limited experiments can improve certain parts of it. An example of this comes from MIT. There is a problem which, in Engineering Faculties in Canada, is usually called the "MIT problem". It can be thrown at students at any age group. Hand them, for example, a brick and ask them to write down things for which it can be used. Usually you get 8 or 9 examples drawn from building practice. I have modified this slightly and give them a telephone relay or some bit of electronic junk and ask them what can be done with it. Of course the maximal marks go to the student who says you can take the wire out and use it to hang the professor for asking a question of this sort. But this idea of convergent and divergent thinking, I think is at the root of Dr. Taylor's remarks.

Turning now to the papers which were presented at the meetings, I notice one significant thing. This is the very small amount of hard data which were presented. A number of people gave us overall plans for grandiose schemes which they have suggested but have not tried. This is fine as long as the schemes are going to be subject to critical appraisal, presumably by a respectable peer group, before they are tried. But I can see, for example, some of the do-gooding international organizations wasting a great deal of do gooding international money implementing schemes which, even at a first look from outside, appear to have numerous holes in them. It would be nice to have a large amount of experience of small systems before indulging in very large ones with the dubious possibility of getting any results. My own view is that the final judgement on CAI will be made on the basis of cost effectiveness. If it costs 25 cents to process one student by conventional methods and costs $25 to do the same thing by computer then it is most unlikely that anybody is going, in the present state of world finance, to put you up the $25. It is nice to do experiments on a limited scale but don't really go away with the rosey-eyed opinion that people are going to put up $25 a time for running these systems in schools. You have in fact to show that some cost benefit which can arise from introducing a CAI system.

This leads naturally to another example of what I conceive to be a valid use for a CAI system. It has been mentioned peripheraly in a number of papers two of which I have heard this morning: Dr. Taylor's paper and another on the computer education of architects. Both authors made the same general point, although I think without singling it out for the specific comment that it deserves, it is that CAI is not important simply because it makes classroom instruction cheaper. I concede the point that the computer is inferior to a human being in almost every way except perhaps as a number cruncher. And even then, those of us who have known people like the late A.C. Aitken, a very well known scottish mathematician and classical scholar and also an arithmetical prodigy, may have some doubts about the arithmetical superiority of computers over the human brain. The point is made by the following story about Aitken. In the 1950's, when I was chairman of the British Computer Society; we invited Aitken to give us a demonstration of his arithmetical powers. It so happened that I had recently obtained from the U.S.A. a printout of the latest computation of pi to 10,000 decimal places. I mentioned this to Aitken when we entertained him to lunch and he said, "Oh, well I want to rest this afternoon, I always rest before these performances. Would you like to lend me this pi printout?" I agreed and he took it with him. In the evening, after he had given one of his phenominal displays of multiplying 10 digit numbers together and giving the answer instantaneously, probably slightly better than a computer could have done at that time because of input - output, he remarked on the fact that we had given him the printout of pi and that he had brought the sheets with him. He said that he would give an example of a thing that a human being could do without being programmed. He handed the sheet to somebody in the front row and invited him to select 5 digits anywhere in the 10,000 digit array. Starting at those 5 digits, Aitken repeated the remainder of digits to the end of pi and came back to the beginning again. I know that it would have taken us much longer to program a computer to do the same thing starting from scratch. So, the computing machine is not likely to displace human beings. I think, however, that the area in which we may look to CAI for advantage is in that of educating bright people. Everybody talks about "disadvantaged" children, this is a word that makes me furious because although it is true that there are a very large number of "disadvantaged" children in our society, it is the unfortunate fact they happen to be the bright children. Again I will speak anecdotically, but I don't think that any experienced educator would dispute the generality of the fact with me. The students who come to colleges of engineering in Canada, (and from experience on faculty at Case Western, I think the same is true in the U.S.A.) are about 4 years "retarded". They come up to the college on this Continent with the knowledge that the European kids have at the age of about 14. The reason for this, in my view at least, is that one has to have a grade 12 content which the mayors or perhaps the prime minister's son can pass. The mayor and the prime minister are often people, of low intellectual though of possibly great political craft, and I don't see why my children should suffer for their stupidity. With a CAI system, one of the points that can be made is that the students can go at their own rate. I am convinced that this is one of the great potential advantages of C.A.I. It has also been pointed out that bright students tolerate extremely bad programming of the system, and this gives a method of system evaluation and of establishing the optimum rate of progress through the system. The difficulty with human teachers is that they have no real desire to accelerate little Johnny because little Johnny is bright. They are much more likely now-a-days to say, well, little Johnny you have done your stuff quickly, suppose you help little Willie and little Bertha. I don't really think this is fair to Johnny.

Let us talk next, very briefly, about technical developments. We saw in one of the early presentations at this symposium a demonstration of a very complete output console. It has many interesting facilities. The touch sensitive tablet developed at NRC is one of them. It forms a beautiful method of direct communication. Anybody who likes to compare it with the light pen, will be impressed by the speed of reaction of the touch sensitive tablet. Two years ago I spent a few weeks at the Lockheed computer centre in Marietta, Georgia, playing with their interactive graphics display. Being of a somewhat impatient disposition, I became utterly frustrated with using the light pen on their CRO and waiting for approximately 15 seconds for any response. This device had a computer which, at that time, was completely dedicated to servicing my terminal. What would have happened if it was servicing 20 others, I don't know. I think that the operators would have voted with their feet. The touch sensitive tablet may give a method of overcoming some of these disadvantages, potentially it seems to have this sort of capability. Having said those nice things about the system which was demonstrated let me remark that a minimum cost for a basic unit of about $2,000 and a cost for a fairly sophisticated unit of $10,000 is far too high even for universities, let alone for schools. I can't really see the Moose Jaw School Board in Saskatchewan ordering 200 of these sets.

You can do a simple sum which shows that this corresponds in cost to the annual operating budget of a fairly large college of engineering. I am not going to say which is the better use of the money. There is another disadvantage to C.A.I., which has not been mentioned anywhere as far as I am aware: If we have large numbers of remote acces stations, we are going to require large numbers of people to keep them in operation. These people do not exist at the moment. Possibly the redundancy of engineers in Canada, and I guess in the United States, and the upsurge of the training of technicians at the TI's will do something to ameliorate the position. But whether or not there be enough people to service the IO equipment in the forseeable future seems to me doubtful. Certainly if devices which in any sense resemble teletypes are used, I would have thought this was very improbable.

The next point I mention is that of acceptability. Dr. Hartley has told us that teachers are very progressive and eager to accept new ideas, but on the other hand Mr. Hutton spent a large amount of his time pointing out that teachers are the most reactionary people. He was being anecdotal when he remarked that, since his own school days, the children had gone from using a slate pencil to using first of all the quill pen. This presumably was in the Montreal region. He then said that, when the steel nib came in there was great difficulty in getting teachers to accept it. They resisted this technical innovation, and their argument was that you can always go into the backyard and pluck a feather from a chicken and thereby make a quill pen but you can't always get a steel nib! One rather wonders to what extent teachers are going to accept CAI when some of the novelty and some of the idea that it may save them work is worn off.

The next area for comment is the psychological basis of teaching. We have heard a lot about educational psychologists and I am sorry to say the impression I got from the papers where they were mentioned was that they were umpopular and that nobody thought too much of them. This is a great pity because, if we don't understand the psychological nature of the process of learning, we are unlikely to do much about having computers improve this. I would maintain that one of the valid uses of CAI is to collect statistics which show the way in which various classes of students go through the learning process. Perhaps even more important we might analyse the group of bright young people who don't too much mind the disorganization of the material presented to them, and use their responses to find better ways of teaching. These are the ways which really active minds devise.

The next area for a very brief comment is that of groups versus single individuals. Now one of the advantages that I have always conceived for CAI is that it allows individuals to go at their own rate, I said something about the implications of this for the gifted a few minutes ago. Of course, the same thing applies to the not so gifted with the additional advantage that the not so gifted are often diffident in participation in open class because they don't want (a) to be thought to be stupid and (b) shown up by the teacher and there are teachers who do this, I am sorry to say. CAI systems will get over part of this difficulty. The question, however, as to whether you use a unit or a group is rather broader. At one time I investigated the teaching of engineering students on a multiple small group basis. One of the techniques was to use extensive video presentation, film and audio tape controlled by a punched paper tape produced by a computer overnight. So although the class started off at the same level in lecture number 1, by the time one week of the course had elapsed, everybody was doing his own thing at a different rate. The tapes were created overnight and used to service the individual groups. This combined some of the advantages of the batch system with some of the advantages of an individual system. To conclude, let me identify what I think to be a few areas of concern. One thing which is completely obvious is that we need curriculum development. I mention this particularly because it is an area in which it is almost impossible to obtain material support. "Curriculum development may lead to the writing a book, and so produce a personal profit", it can't be supported from national funds. I think these are fallacies. If we don't have curriculum development we shall not have any progress in CAI. Have we any areas of success? I identify two from the presentations which have been made here. The first of them is in using a computer to teach students programming languages. This is outstandingly successful, practically all university level students are exposed to a situation in which the computer teaches them to program. APL is an obvious example on one of the languages which is used. Basic is another one, Fortran yet a third. I view programming languages, whether they are CAI languages or number crunching languages, as being meritorious in proportion to the extent to which the student doesn't need teaching. I like Basic for the reason that, after the first batch of our students was exposed to basic, we were able to discontinue all formal instruction. It is such a simple language that students learn by looking over the shoulder of one of their colleagues who is running a program on the terminal. That is the sort of language that we want, not the sort of language where the rules really foul you up. A second area of success is in the use of computers in simulation. This again is something which we have seen in a number of the papers presented at this meeting: the simulation of games of business are quite old, they formed one of the earliest non-numerical applications of computing machines. A more modern simulation is to use computing machines to provide, simulated experiments for students. We are using this in Saskatchewan in the electrical power systems laboratory where we have a simulation for the provincial power network. This gives the students the advantage of being able to make stupid mistakes on a large system which, if committed on the real system might lead to a power outage over half of the province of Saskatchewan. In view of the interconnections which were involved these days, it might even put out the lights in Boston and perhaps, as we saw on another occasion, increase the birth rate!

What else does Canada need? For one thing we need a standard language. Despite the remarks we heard earlier on the undesireability of identifying one part of the system, Canada is in a state of being an undeveloped country in this area. If we let Canada become a developed country, then it's too late. So why not do some thinking now? To whitewash ourselves, so that MIT doesn't get a false view of Canada in this respect, let me just remark that one of the subjects of agreement at the Associate Committee meeting on Tuesday was that, in developing a Canadian standard language, we must make use of all of the facilities which we have got in current languages. We should select their best features and merely add those special lectures which we need. We don't want to write something from scratch. Finally do we need CAI? I am going to suggest to you that by far the most efficient medium of instruction is the book. Perhaps if we taught students to read at an early age, there mightn't be a need for CAI except for remedial treatment.